Vol.2021-DPS-189 No.11
2021/12/21

秘匿分解データを用いた新しい機械学習

宮島 洋文[1,a)]　重井 徳貴[2,b)]　宮島 廣美[2,c)]　白鳥 則郎[3,d)]

概要：持続可能な社会の実現に向けて, SDGs (Sustainable Development Goals) に対する取り組みが世界中で模索されている. 日本では Society 5.0 が目指す超スマート社会の構築をテーマの一つとして掲げている. 超スマート社会は, サイバー空間とフィジカル空間（現実社会）の高度な融和を目指すものであり, ビッグデータを AI が解析することで, 個人のニーズに合った有効な情報がより迅速に現実社会にもたらされる. それでは, ビッグデータに対するプライバシーの侵害や管理強化を防ぐ超スマート社会はどのように構築すれば良いのであろうか. この問題の解決のためには, サイバー空間でのビックデータのプライバシーを保護する AI の解析手法の開発が重要になる. これまでに, この分野では, ユーザにとって安心・安全な AI の解析法としての機械学習の開発の観点から, 1）秘密共有+ SMC (Secure Multiparty Computation), 2）準同型暗号化, 3）連合学習, 等に関する研究が行われているが, 学習法の利活用に対する価値とプライバシーのリスクに対するバランスが高度にとれた方法は知られていない.

これらの背景を踏まえて, 本論文では, 簡易秘匿分解データを用いた分散処理による学習法を提案する. この方法では, あらかじめ個々のデータを乱数を使って複数に分解し, それぞれの断片を各サーバに保存する. 学習は断片データを使って, 各サーバでの部分計算と中央サーバでの統合計算を繰り返し実行する. 提案法の利点としては, 学習データをそのまま使うことがないことによるセキュリティの向上と, 連合学習と同様に分散処理による機械学習法の実現により多くの問題への利活用が容易となる. この提案法に基づいて, 機械学習の応用例として, 簡易秘匿分解データを用いた分散処理による BP (Back Propagation) のアルゴリズムを提案し, その有効性を示す.

New Machine Learning Method with using Secure Divided Data

1. はじめに

1.1 SDGs から超スマート社会へ

持続可能な社会の実現に向けて, 17 のグローバル目標と 169 の達成基準から構成される SDGs (Sustainable Development Goals) に対する取り組みが模索されている [1], [2].

このような SDGs のガイドラインに沿った持続可能な社会の産業面からの促進に向けて, 日本では Society 5.0 をテーマの一つとして掲げている. Society 5.0 が目指す超スマート社会では, サイバー空間とフィジカル空間（現実社会）が高度に融和し, サイバー空間に存在する人工知能（AI）が状況に応じて必要な情報を瞬時に見つけだし, その分析結果を現実社会に提供する [3], [4]. Society 5.0 では, 現実社会のセンサーやスマートフォンなどの IoT デバイスなどからの膨大な情報がサイバー空間に蓄積される. サイバー空間では, このビッグデータを AI が解析することで, 個人のニーズに合った有効な情報が, より迅速に現実社会にもたらされる.

このような超スマート社会では,「快適な生活」,「健康な生活の促進」,「高齢者の自立支援」に対する大いなる貢献が期待される一方で,「AI からの情報の説明能力不足」,「プライバシーの侵害」や「個人情報の管理強化」がデメリットとなる [5], [6], [7]. それでは, このようなデメリットの一つであるプライバシーの侵害や管理強化を防ぐ超スマート社会はどのように構築すれば良いのであろうか.

この問題の解決のためには, サイバー空間でのビックデー

[1] 長崎大学
Nagasaki University, Nagasaki 852–8521, Japan
[2] 鹿児島大学
Kagoshima University
[3] 中央大学
Chuo University
[a)] miyajima@nagasaki-u.ac.jp
[b)] shigei@eee.kagoshima-u-ac.jp
[c)] k2356323@kadai.jp
[d)] norio@shiratori.riec.tohoku.ac.jp

ⓒ 2021 Information Processing Society of Japan

タのプライバシーを保護する AI の解析手法を開発することが重要になる. 本論文では, この手法を高度 AI 処理と呼ぶ. それゆえ, この分野では, ユーザにとって安心・安全を保ちつつ機械学習を行う高度 AI 手法に関する研究が活発に行われている [2], [3], [4], [5], [6], [7].

1.2 プライバシー保護の社会の実現に向けて

高度 AI 処理を実行するインフラであるクラウドまたはエッジシステムでのビッグデータ処理については, データの秘匿性を保ちながら計算処理を行う技術の開発が目標の一つとなる. 機械学習のためのデータのプライバシー保護の観点からは, 1) 秘密共有+ SMC (Secure Multiparty Computation) [8], [9], [10], 2) 準同型暗号化 [11], [12], 3) 連合学習 (Federated Learning : FL) [13], [14] 等に関する研究が行われている.

1) の方法は, 最初に各データに乱数を使って複数の断片に分割して, それぞれを各サーバに保存する. 次に, 各々のサーバが持つデータの断片を使って, それらの秘匿性を保ちつつサーバ間で協調計算を実現する. データは, 分解されたまま秘匿性の保存される演算のみを使って計算されるので, プライバシーは保護される [8], [9], [10].

2) の方法は, はじめに各データの暗号化を行う. 次に, 暗号化されたデータを使って所望の演算を行い, 最終的に得られた結果を復号化する事で計算が実行される. この方法では, データを暗号化したまま計算が可能となるような暗号化の方法を見つけることが目標となる [11], [12].

3) の方法は, 1 つの中央サーバと複数のサーバから構成される. はじめにデータ集合は複数の部分集合に分かれており, 各々は各サーバに保存される. この部分集合を使って各サーバで機械学習の計算を独立に行い, 得られた結果を中央サーバに送る. 中央サーバでは, これらを統合して, すべてのデータに対する結果を計算し, その結果を各サーバに送る. 以下, 各サーバでの部分結果と中央サーバでの全体の結果を求める計算過程を繰り返す. この過程で, 各サーバのデータはサーバ内部のみの計算に用いられるだけで, 外部に出ていくことはない. このことにより, データの秘匿性が保証される [13], [14].

これらの場合, 1) と 2) の方法はデータの暗号化や乱数を使ってプライバシーを厳密に保護する方法であり, 3) は全データを部分集合に分けて各サーバに分散し, データを各サーバの外部に取り出すことなく分散処理によって学習を実行する方法である. それぞれに長所と短所がある. 1) と 2) の方法は, セキュリティについては, 極めて秘匿性の高い方法であるが, 機械学習への利活用性については, 応用が限られる. 3) の方法は, 手続きの単純さから機械学習の多くの問題への利活用性は高いと言える. また, IoT (Internet of Things) 向きのエッジコンピューティングへの適応性も高い. 一方で, セキュリティレベルは, 1) と 2) の方法に比べると低い. それゆえ, この分野の研究は, 学習法の利活用の価値とセキュリティのリスクのバランスを高めた学習法の構築が目標となる.

1.3 秘匿性と実用性の高い機械学習の提案

本論文では, 簡易秘匿分解データを用いた分散処理による第 4 の学習法を提案する.

この方法では, あらかじめ個々のデータを 1) の方法と同様に乱数を使って複数に分解し, それぞれのサーバに保存する. 学習は分解されたデータを使って, 各サーバごとの独立した計算と中央サーバでの統合計算を繰り返し実行することにより実現される. それゆえ, この方法では, 多くの問題への利活用が容易である. また, 3) の方法のように各サーバ内においてデータそのものを使って学習することがなく, 分解されたデータを使った秘匿性の高い学習が可能となる. これらの特徴から, 提案法は利活用の価値が高くセキュリティリスクの低い方法として期待できる.

機械学習の目的は, 与えられたデータから, データ間の関係を結びつけるパラメータを推定することである. 機械学習には, 入力と出力データ間の関係を学習する教師あり学習と, データの分布を近似する教師なし学習がある.

本論文では, 機械学習の例として, 簡易秘匿分解データを用いた分散処理による BP のアルゴリズムを提案する. 本論文の内容は以下のようになる.

2 章では, BP 法に関する簡易秘匿データとして加算や積型の分解データを定義する. 3 章では, 各サーバに分散された簡易秘匿分解データを使って分散処理による学習法を提案する. 4 章では, 従来型の BP と提案法の BP 法の精度を数値実験により比較する. 5 章ではまとめとして本研究の社会への貢献と今後の展望を述べる.

2. 予備概念

本章では, はじめに, 分解したデータによる分散処理の計算法の概念を説明する. 次に, 提案法で用いるデータの分解法について説明する. さらに, 最急降下法を導入し, これを用いた BP 法について説明する.

2.1 簡易秘密計算法の概略

本論文で用いるモデルとして簡易秘匿分解データを用いる分散処理計算法について説明する. 図 1 のような L 個の端末と $Q+1$ 個のサーバからなる分散処理方式のモデルを使って, 実数データ x の関数 $f(x)$ を計算する場合について説明する.

はじめに, 1 つの端末から与えられた実数データ x を Q 個のランダムな実数に分解する. q 番目のサーバには, データ x の断片である x^q が送られる. ここに, $x = \sum_{q=1}^{Q} x^q$ とする. q 番目のサーバで $f_q(x^q)$ を計算し, サーバ 0 に結果を送る. ここに, $f_q(\cdot)$ は q 番目のサーバが行う計算処理とする. サーバ 0 では, 各サーバでの計算結果 $f_q(x^q)$ を統合して最終結果 $f(x) = \odot_{q=1}^{Q} f_q(x^q)$ を得る. ここに, $\odot_{q=1}^{Q}$

は統合計算を示す. 1 回の処理で結果が得られない場合は, サーバ 0 の結果 $f(x)$ が各サーバに送られて, 同様の過程が繰り返される.

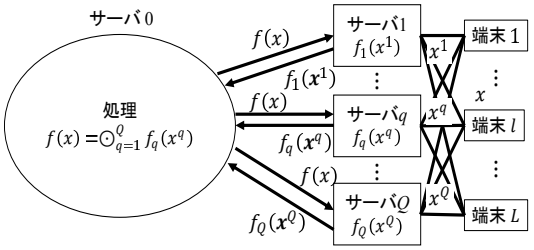

図 1 簡易秘匿分解データによる計算モデル

2.2 提案法によるデータ分割法

データを格納するサーバの個数は 2, すなわちデータを 2 つの断片に分解する場合について説明する.

実数値 a, b をそれぞれ $a = a^{(1)} + a^{(2)} = A^{(1)}A^{(2)}$, $b = b^{(1)} + b^{(2)} = B^{(1)}B^{(2)}$ と和と積の形で分割する. ここで, $a^{(1)}$, $a^{(2)}$, $A^{(1)}$, $A^{(2)}$, $b^{(1)}$, $b^{(2)}$, $B^{(1)}$, $B^{(2)}$ はランダムに選ばれた実数値であり, $a^{(1)}$, $b^{(1)}$, $A^{(1)}$, $B^{(1)}$ は Server 1, $a^{(2)}$, $b^{(2)}$, $A^{(2)}$, $B^{(2)}$ は Server 2 に保存される. 例えば, 実数値 a, b は以下のように分割される.

$a = a^{(1)} + a^{(2)}$: $a^{(1)} = a(r_1/10)$, $a^{(2)} = a(1 - r_1/10)$
$b = b^{(1)} + b^{(2)}$: $b^{(1)} = b(r_1/10)$, $b^{(2)} = b(1 - r_1/10)$
$a = A^{(1)}A^{(2)}$: $A^{(1)} = \sqrt{a}(r_2/10)$, $A^{(2)} = \sqrt{a}(10/r_2)$
$b = B^{(1)}B^{(2)}$: $B^{(1)} = \sqrt{b}(r_2/10)$, $B^{(2)} = \sqrt{b}(10/r_2)$

ここで, r_1 と r_2 はそれぞれ $-9 \leq r_1 \leq 9$, $r_1 \neq 1$, $0.2 \leq r_2 \leq 9$, $r_2 \neq 1$ を満たす, ランダムに選ばれた実数値とする.

例えば, データ ID=1 について, $a^{(1)} = 50\times(4/10) = 20$, $a^{(2)} = 50\times(1 - 4/10) = 30$, $A^{(1)} = \sqrt{50}\times(9/10) = 6.31$, $A^{(2)} = \sqrt{50}\times(10/9) = 7.86$ となる.

ここで, 科目 A の総和および平均値の計算について説明する. Server 1 では項目 $a^{(1)}$, Server 2 では項目 $a^{(2)}$ の和を求める. この例の場合は, 項目 $a^{(1)}$ の和はマイナス 23, 項目 $a^{(2)}$ の和は 328 となる. これらの和を求めることで, a の総和 $-23 + 328 = 305$ を求めることができる. 同様にして, $a^{(1)}$ の平均-4.6 と $a^{(2)}$ の平均 65.6 を用いることで, a の平均値 $-4.6 + 65.6 = 61$ を求めることができる. この場合, 各データはランダムな数値を用いて分解されており, サーバは各データの断片を用いて計算を行い, 結果を統合することにより計算を実行できる.

さらに, このような加算や乗算型のデータ分解法を用いると, 以下の関係が成り立つ.

1) $a + b = (a^{(1)} + b^{(1)}) + (a^{(2)} + b^{(2)})$
2) $a - b = (a^{(1)} - b^{(1)}) + (a^{(2)} - b^{(2)})$
3) $ab = (A^{(1)}B^{(1)})(A^{(2)}B^{(2)})$
4) $a/b = (A^{(1)}/B^{(1)})(A^{(2)}/B^{(2)})$

これらを用いることで, ランダムに選ばれた実数値 $a^{(1)}$, $a^{(2)}$, $A^{(1)}$, $A^{(2)}$, $b^{(1)}$, $b^{(2)}$, $B^{(1)}$, $B^{(2)}$ のみを用いて, a, b を復号化することなく, 各サーバでの計算結果を使って, a と b に対する四則演算を実行できる.

2.3 階層型ニューラルネットワークと BP 法

SDM に基づく教師あり学習の 1 つである BP 法について説明する. ここでは, 一般性を失うことなく 3 層の階層型ニューラルネットワークに対する BP 学習について説明する [20], [22].

ニューラルネットワークにより構成される写像 $\boldsymbol{h}$: $J_{in}^n \to J_{out}^R$ を, $\boldsymbol{x} \in J_{in}^n$ に対して $\boldsymbol{h}(\boldsymbol{x}) = (h_1(\boldsymbol{x}), \cdots, h_R(\boldsymbol{x}))$ と定義する. ただし, $J_{in} = [0,1]$ または $[-1,1]$, $J_{out} = \{0,1\}$ とする. この場合, L 個の学習データの集合 $X = \{(\boldsymbol{x}^l, \boldsymbol{d}(\boldsymbol{x}^l)) | \boldsymbol{x}^l \in J_{in}^n, \boldsymbol{d}(\boldsymbol{x}^l) \in J_{out}^R, l \in Z_L\}$ を使って, ネットワークのパラメータである重みを決定する. ここに, $\boldsymbol{d}(\boldsymbol{x}^l) = \big(d_1(\boldsymbol{x}^l), \ldots, d_R(\boldsymbol{x}^l)\big)$ は入力 $\boldsymbol{x}^l$ に対する (教師データの) 出力をあらわす.

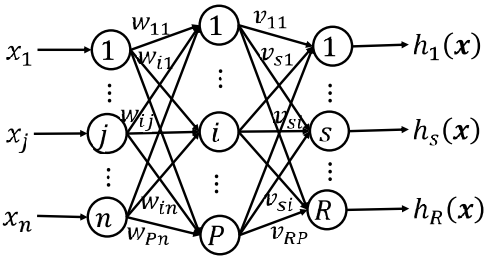

図 2 3 層階層型ニューラルネットワークの例

ネットワークの重みを $W = \{w_{ij} | i \in Z_P, j \in Z_n^*\}$, $V = \{v_{si} | s \in Z_R, i \in Z_P^*\}$ とする. このとき, ネットワークの出力は次式により得られる.

$$y_i(\boldsymbol{x}) = \frac{1}{1 + \exp\left(-\left(\sum_{j=0}^{n} w_{ij} x_j\right)\right)} \tag{1}$$

$$h_s(\boldsymbol{x}) = \frac{1}{1 + \exp\left(-\left(\sum_{i=0}^{P} v_{si} y_i(\boldsymbol{x})\right)\right)} \tag{2}$$

ここに, $x_0 = 1$, $y_0 = 1$ であり, w_{i0}, v_{s0} は各層のしきい値をあらわす.

学習の評価関数を式 (3) として与える.

$$E(X, W, V) = \frac{1}{2L} \sum_{l=1}^{L} \sum_{s=1}^{R} \big(d_s(\boldsymbol{x}^l) - h_s(\boldsymbol{x}^l)\big)^2 \tag{3}$$

重み W, V は, 与えられた学習用データ X に対する式 (3) の最小化問題として BP 法により解くことができる. BP 法のアルゴリズムを以下のように与える. X は学習用データ, $B(\subset X)$ は学習用データの部分集合, T は重みの最大更新回数, θ はしきい値, α は学習係数とする.

[**B**P 法のアルゴリズム] BP(X, W, V) [22]
入力 : 学習用データ
$X = \{(\boldsymbol{x}^l, \boldsymbol{d}(\boldsymbol{x}^l)) | \boldsymbol{x}^l \in J_{in}^n, \boldsymbol{d}(\boldsymbol{x}^l) \in J_{out}^R, l \in Z_L\}$
出力 : 重み $W = \{w_{ij} | i \in Z_P, j \in Z_n^*\}$,
$V = \{v_{si} | s \in Z_R, i \in Z_P^*\}$
[**S**tep 1]
W, V を初期化, $t \leftarrow 0$ とする.
[**S**tep 2]

表 1　簡易秘匿分解データを用いた計算例.

ID	subject A	subject B	Additional form					Multiplication form				
				a		b			A		B	
	a	b	r_1	$a^{(1)}$	$a^{(2)}$	$b^{(1)}$	$b^{(2)}$	r_2	$A^{(1)}$	$A^{(2)}$	$B^{(1)}$	$B^{(2)}$
1	50	80	4	20	30	32	48	9	6.31	7.86	8.05	9.94
2	40	50	-6	-24	64	-30	80	2	1.27	31.62	1.41	35.36
3	65	30	2	13	52	6	24	0.8	0.65	100.78	0.44	68.47
4	70	62	-8	-56	126	-49.6	111.6	5	4.18	16.73	3.94	15.75
5	80	40	3	24	56	12	28	4	3.58	22.36	2.53	15.81
sum	305	262		-23	328	-29.6	291.6					
average	61	52.4		-4.6	65.6	-5.92	58.32					

学習用データ X の部分集合 $B \subset X$ をランダムに選択する. 各データ $(\boldsymbol{x}^l, \boldsymbol{d}(\boldsymbol{x}^l)) \in B$ に対して, 式 (1), (2) により $y_i(\boldsymbol{x}^l)$, $h_s(\boldsymbol{x}^l)$ を求める.

[**S**tep 3]

次式に基づき $w_{ij} \in W$, $v_{si} \in V$ を更新する.

$$w_{ij} \leftarrow w_{ij} + \alpha \sum_{\boldsymbol{x}^l \in B} \sum_{s=1}^{S} (d_s(\boldsymbol{x}^l) - h_s(\boldsymbol{x}^l))(1 - h_s(\boldsymbol{x}^l))v_{si} \times y_i(\boldsymbol{x}^l)(1 - y_i(\boldsymbol{x}^l))x_j^l \quad (4)$$

$$v_{si} \leftarrow v_{si} + \alpha \sum_{\boldsymbol{x}^l \in B} (d_s(\boldsymbol{x}^l) - h_s(\boldsymbol{x}^l))h_s(\boldsymbol{x}^l) \times (1 - h_s(\boldsymbol{x}^l))y_i(\boldsymbol{x}^l) \quad (5)$$

[**S**tep 4]

式 (3) に基づき評価値 $E(X, W, V)$ を求める.

[**S**tep 5]

もし, $E < \theta$ または $t < T$ ならば, アルゴリズムを終了する. そうでなければ, $t \leftarrow t+1$ として Step 2 へ. □

特に, Step 2 において $|B| = 1$ のときオンライン学習, $|B| = |X|$ のときバッチ学習, $1 < |B| < |X|$ のときミニバッチ学習とよばれる.

3. 簡易秘匿分解データによる学習法

本章では, プライバシーを保護する第 4 の方法として, 簡易秘匿分解データを用いる分散処理の学習法を提案する. 3.1 節では, このようなパラメータの更新法が機械学習として成立する. すなわち, 分解データによる分解パラメータの部分的な更新の統合が, 全体の誤差 (目的関数) を減らす方向に進んでいくことを示す. 3.2 と 3.3 節では, この学習に基づく BP 学習法を提案する.

3.1 分解データによる最急降下法

提案法は, あらかじめ学習データとパラメータを分解し, それぞれを各サーバに記憶する. それらを用いて各サーバで更新したパラメータの結果を中央サーバで統合することを繰り返すことにより学習を実行する. それでは, このように分解したパラメータを更新して統合することによる最急降下法は, 全体としてうまく働くであろうか. すなわち, 全体の誤差は学習が進むにつれて下がる方向に動くであろうか. 最急降下法のパラメータの更新式は一般に以下のように, 与えられる [21].

$$\frac{d\theta_i}{dt} = -\frac{\partial E}{\partial \theta_i} \text{ for } i = 1, \cdots, m \quad (6)$$

ここに, E は目的関数, m はパラメータ数である.

いまこのパラメータが Q 個の分解した成分からなるとする. すなわち,

$$\theta_i = f(\theta_i^{(1)}, \cdots, \theta_i^{(Q)}) \text{ for } i = 1, \cdots, m \quad (7)$$

とする. このモデルに対する分解したパラメータの更新式を以下のように仮定する.

$$\frac{d\theta_i^{(q)}}{dt} = -\frac{\partial E}{\partial \theta_i^{(q)}} \quad (8)$$
$$i = 1, \cdots, n, \ q = 1, \cdots, Q$$

このとき, 以下の関係が成立する.

$$\frac{dE}{dt} \leq 0 \quad (9)$$

等式は式 (8) の右辺が 0 のとき成立する.

なぜなら, 式 (9) の左辺は以下のように書き換えられる.

$$\frac{dE}{dt} = \sum_{i,q} \frac{\partial E}{\partial \theta_i} \frac{\partial \theta_i}{\partial \theta_i^{(q)}} \frac{d\theta_i^{(q)}}{dt} \quad (10)$$

式 (8) を以下のように書き換えて

$$\frac{d\theta_i^{(q)}}{dt} = -\frac{\partial E}{\partial \theta_i} \frac{\partial \theta_i}{\partial \theta_i^{(q)}} \quad (11)$$
$$q = 1, \cdots, Q$$

これを式 (10) に代入すると

$$\frac{dE}{dt} = -\sum_{i,q} \left(\frac{\partial E}{\partial \theta_i} \times \frac{\partial \theta_i}{\partial \theta_i^{(q)}} \right)^2 \leq 0 \quad (12)$$

この結果は, 各サーバ上での分解されたパラメータの更新 (式 (8)) が, 全体の誤差の減少 (式 (12)) に導くことを示している.

3.2 秘匿分解データによる BP 学習

本節では, はじめに BP 学習で用いるデータやパラメータの分解法, すなわち式 (7) の関係を導入する. 学習用データ $(\boldsymbol{x}^l, \boldsymbol{d}(\boldsymbol{x}^l)) \in X$ は以下のように Q 個にランダムに分割され, Q 個のサーバに分けて管理されると仮定する [19], [20].

$$x_j^l = \Pi_{q=1}^{Q} x_j^{l(q)} \quad (13)$$

$$d_s(\boldsymbol{x}^l) = \sum_{q=1}^{Q} d_s^{(q)}(\boldsymbol{x}^l) \tag{14}$$

また, 重み $w_{ij}\in W$, $v_{si}\in V$ もまた以下のように Q 個に分割され, Q 個のサーバに分けて記憶されるものとする.

$$w_{ij} = \Pi_{q=1}^{Q} w_{ij}^{(q)} \tag{15}$$

$$v_{si} = \Pi_{q=1}^{Q} v_{si}^{(q)} \tag{16}$$

このとき, 分割された重みの成分の集合を $W^{(q)} = \{w_{ij}^{(q)} | i\in Z_P, j\in Z_n^*\}$, $V^{(q)} = \{v_{si}^{(q)} | s\in Z_S, i\in Z_P^*\}$ とする.

ここで, $\Pi_{q=1}^{Q} x_0^{(q)} = 1$ とする.

式 (13) により分割された入力データ対する階層型ニューラルネットワークの出力は, 式 (1), (2) の代わりに次式 (17), (18) により導出することができる.

$$y_i(\boldsymbol{x}) = \frac{1}{1+\exp\left(-\left(\sum_{j=0}^{n} \Pi_{q=1}^{Q} w_{ij}^{k(q)} x_j^{(q)}\right)\right)} \tag{17}$$

y_i を計算後, $y_i = \Pi_{q=1}^{Q} y_i^{(q)} (i\in Z_P^*)$ と分割し, 各サーバに送る. ここで, $\Pi_{q=1}^{Q} y_0^{(q)} = 1$ とする.

各サーバで重みを乗じてサーバ 0 で以下の $h_s(\boldsymbol{x})$ を計算する.

$$h_s(\boldsymbol{x}) = \frac{1}{1+\exp\left(-\left(\sum_{i=0}^{P} \Pi_{q=1}^{Q} v_{si}^{k(q)} y_i^{(q)}\right)\right)} \tag{18}$$

その後, $h_s(\boldsymbol{x}) = \sum_{q=1}^{Q} h_s^{(q)}(\boldsymbol{x})$ と分割し, 各サーバに送る.

このとき, データ集合 X に対する平均二乗誤差は次式により求められる.

$$E(X) = \frac{1}{2L} \sum_{l=1}^{L} \sum_{s=1}^{S} \left(\sum_{q=1}^{Q} (d_s^{(q)}(\boldsymbol{x}^l) - h_s^{(q)}(\boldsymbol{x}^l)) \right)^2 \tag{19}$$

BP 法については, 式 (4), (5) の代わりに式 (8) に基づいて式 (20), (21) を用いることで, 分割された重み $w_{ij}^{(q)} (i\in Z_P, j\in Z_P^*)$, $v_{si}^{(q)} (s\in Z_S, i\in Z_P^*)$ を更新することができる.

$$\begin{aligned} w_{ij}^{(q)} = w_{ij}^{(q)} + \ & \alpha \sum_{\boldsymbol{x}^l\in B} \sum_{s=1}^{S} \left(\sum_{q=1}^{Q} (d_s^{(q)}(\boldsymbol{x}^l) - h_s^{(q)}(\boldsymbol{x}^l)) \right) \\ & \times (1 - h_s(\boldsymbol{x}^l)) \Pi_{q=1}^{Q} v_{si}^{(q)} y_i^{(q)}(\boldsymbol{x}^l)(1 - y_i(\boldsymbol{x}^l)) \\ & \times (\Pi_{q=1}^{Q} w_{ij}^{(q)} x_j^{l(q)}) / w_{ij}^{(q)} \end{aligned} \tag{20}$$

$$\begin{aligned} v_{si}^{(q)} = v_{si}^{(q)} + \alpha & \sum_{\boldsymbol{x}^l\in B} \sum_{q=1}^{Q} (d_s^{(q)}(\boldsymbol{x}^l) - h_s^{(q)}(\boldsymbol{x}^l)) h_s(\boldsymbol{x}^l) \\ & \times (1 - h_s(\boldsymbol{x}^l)) (\Pi_{q=1}^{Q} y_i^{(q)} v_{si}^{(q)}) / v_{si}^{(q)} \end{aligned} \tag{21}$$

ここに, 式 (20) と (21) の右辺の更新式において $1/w_{ij}^{(q)}$ と $1/v_{si}^{(q)}$ に注意する.

ここで示した分解データと分散処理に基づく学習法の概略を以下に示す. なお, ここでは, 各パラメータ $p\in P$ は, 積演算を用いて $p = \Pi_{q=1}^{Q} p_q$ と断片化されているものとする.

[提案学習法の概略]

入力 : データ集合 X

出力 : 学習パラメータの集合 P

[**S**tep 1]

全てのデータとパラメータを秘密分散を用いて断片化, 無意味化を行い, それぞれのサーバ q に送信する. サーバに送信されたデータの集合を X_q, パラメータの集合を P_q とする.

[**S**tep 2]

パラメータの更新に使用するデータの集合 $B\subseteq X$ を選択する. ここで, B を断片化したデータの集合を $B_1, \cdots, B_Q$ とおく.

[**S**tep 3]

サーバ q において, $g_q = F(p_q, B_q)(p_q\in Pq)$ を計算し, サーバ 0 に送る.

[**S**tep 4]

サーバ 0 で g_q を統合し, パラメータ p に対する更新量 $\triangle p$ を求め, 各サーバに送信する.

[**S**tep 5]

各サーバにおいて, 次式により各パラメータ $p_q\in P_q$ を更新する.

$$p_q \leftarrow p_q + \triangle p / p_q \tag{22}$$

[**S**tep 6]

終了条件を満たしていれば終了, 満たさなければ Step 2 へ. □

表 2 はこの流れに基づいた, BP 法のアルゴリズムである. $|B| = 1$ の場合はオンライン学習, $|B| = |X|$ の場合はバッチ学習, $|B| < |X|$ の場合はミニバッチ学習となる. Step 2 で各サーバにおいて重みと入力の積を計算し, Step 3 で Step 2 の結果を統合して中間層の出力を計算し, 結果を分解して各サーバに送る. Step 4 で重みと中間層の出力積を計算し, Step 5 でこの結果を統合して出力層の結果を分解して各サーバに送る. Step 5 で各サーバのもつ学習データの断片と Step 6 で送られてきた出力層の結果の断片の差を計算しサーバ 0 に送る. Step 7 では, Step 6 で求めた断片の差を使って, 誤差の計算を行い, 学習終了かどうかの判定を行う. 終了でなければ, 次の学習で用いるデータの番号の部分集合 U を決定する. さらに, 部分集合 U に対応する学習データに対する誤差を計算し, 更新量を計算し, 各サーバに送る. Step 8 において, 各サーバのもつパラメータの断片を更新する. なお, Step 6 の定数 a は, サーバ 0 に対して出力データを秘匿するために使っている.

表 2 においては, MSE の計算を行う際に分解重みの更新量の計算も行うことで (Step 7), NN の出力計算の回数を削減し, サーバ間の通信回数を抑えている.

4. 数値実験

ここでは, 表 3 に示すような Iris, Wine, Sonar, BCW, Spam の 5 種類のデータに対して, 従来法と提案法のアルゴリズムによるデータの分類を行う [24]. #data:L はデータ数が L 個であることを意味する.

表 2　簡易秘匿分解データに対する BP 法 (オンライン学習, バッチ学習, ミニバッチ学習)

	サーバ 0	サーバ q (q∈Z_Q)
初期化		$\{x_j^{l(q)}\|l\in Z_L, j\in Z_n\}$, $\{d_s^{(q)}(\boldsymbol{x}^l)\|l\in Z_L, s\in Z_R\}$ を記憶 $\{w_{ij}^{(q)}\|i\in Z_P, j\in Z_n^*\}$, $\{v_{si}^{(q)}\|s\in Z_R, i\in Z_P^*\}$ を初期化
Step 1	t←0	
Step 2		$w_{ij}^{(q)}x_j^{l(q)}(\boldsymbol{x}^l, i\in Z_P, j\in Z_n^*)$ を計算し, サーバ 0 に送る.
Step 3	式 (17) により中間層の出力 $y_i(\boldsymbol{x}^l)$ を計算し, $y_i=\Pi_{q=1}^Q y_i^{(q)}$ と分割する. $y_i^{(q)}(q\in Z_Q)$ を各サーバに送る.	
Step 4		教師データと NN の出力の差の成分 $v_{si}^{(q)}y_i^{(q)}(s\in Z_R, i\in Z_P^*)$ を計算し, サーバ 0 に送る.
Step 5	式 (18) により NN の出力 $h_s(\boldsymbol{x}^l)(s\in Z_S)$ を求める. $h_s(\boldsymbol{x}^l)=\Pi_{q=1}^Q h_s^{(q)}(\boldsymbol{x}^l)$ と分割する. $h_s^{(q)}(\boldsymbol{x}^l)(q\in Z_Q)$ を各サーバに送る.	
Step 6		$a\left(d_s^{(q)}(\boldsymbol{x}^l)-h_s^{(q)}(\boldsymbol{x}^l)\right)(s\in Z_R)$ を計算し, サーバ 0 に送る.
Step 7	式 (19) に基づき教師データと NN の出力の誤差 $E(X,W,V)$ を求める. $E<\theta$ または t≦T の場合は学習終了. それ以外の場合は, 自然数の集合 U⊆Z_L をランダムに選択し, 各サーバの出力の誤差の成分を統合して更新量 $p_{1(ij)}$, $p_{2(si)}$ を計算する. $p_{1(ij)}=\sum_{l\in U}\sum_{s=1}^R\sum_{q=1}^Q a(d_s^{(q)}(\boldsymbol{x}^l)-h_s^{(q)}(\boldsymbol{x}^l))$ $\times(1-h_s(\boldsymbol{x}^l))v_{si}y_i(\boldsymbol{x}^l)(1-y_i(\boldsymbol{x}^l))(\Pi_{q=1}^Q w_{ij}^{(q)}x_j^{l(q)})$, $p_{2(si)}=\sum_{l\in U}\sum_{q=1}^Q a(d_s^{(q)}(\boldsymbol{x}^l)-h_s^{(q)}(\boldsymbol{x}^l))h_s(\boldsymbol{x}^l)$ $\times(1-h_s(\boldsymbol{x}^l))(\Pi_{q=1}^Q y_i^{(q)}v_{si}^{(q)})$ を計算し, 各サーバに送る.	
Step 8		更新量 $p_{1(ij)}$, $p_{2(si)}$ を用いて重みの各成分 $\{w_{ij}^{(q)}\|i\in Z_P, j\in Z_j^*\}$, $\{v_{si}^{(q)}\|s\in Z_R, i\in Z_P^*\}$ を更新する. $w_{ij}^{(q)}$←$w_{ij}^{(q)}+\alpha p_{1(ij)}/aw_{ij}^{(q)}$ $v_{si}^{(q)}$←$v_{si}^{(q)}+\alpha p_{2(i)}/av_{si}^{(q)}$ t←$t+1$ として Step 2 へ.

表 3　数値実験で扱うデータ集合

	Iris	Wine	Sonar	BCW	Spam
#data : L	150	178	208	683	4601
#input : n	4	13	60	9	57
#output : R	3	3	2	2	2

ここでは, 式 (1), (2) において $P=10$, Iris と Wine の場合は $S=3$, Sonar と BCW と Spam の場合は $S=2$ とする 3 層ニューラルネットワークを用いた. 提案手法においては $Q=3$, $a=1$ とした. なお, ここでは 5-fold cross-validation による評価を行う. 実験の条件としては, 最大学習回数を $T=50000$, 学習係数を $K_w=0.01$, $K_v=0.01$ とする. 実験においては, 学習用データに対する平均二乗誤差 (Mean Square Error : MSE) がしきい値 θ を下回る, または重みの更新回数が最大学習回数となれば学習終了とする. なお, しきい値は Iris, Wine については $\theta=3.0\times10^{-2}$, Sonar, BCW については $\theta=4.0\times10^{-2}$, Spam については $\theta=1.0\times10^{-1}$ を用いた.

学習終了後, 各手法について, 学習用データおよびテスト用データに対する誤分類率を比較する.

オンライン学習, バッチ学習, ミニバッチ学習に関する実験の結果を表 4 に示す. ここで, A1 は通常のオンライン BP 法, A2 は通常のバッチ法, A3 は通常のミニバッチ BP 法 [22], B1 は Horizontally Partitioned Data (HPD) を用いたオンライン BP 法, B2 は HPD におけるバッチ BP 法, B3 は HPD におけるミニバッチ BP 法 [13], C1 は提案法によるオンライン BP 法, C2 は提案法によるバッチ BP 法, C3 は提案法によるミニバッチ BP 法を示す. ミニバッチ BP 法においては, 学習用データのうち 1/3 個をランダムに選択して重みの更新を行うものとする. また, Learn, Test はそれぞれ学習用データおよびテスト用データに対する誤分類率 (%), LT は学習終了時点における重みの更新回数を意味する. 表中の値はそれぞれ 20 回試行の平均値である.

表 4 に示す結果より, 提案手法は従来の BP 法, および従来の BP 法と組み合わせた BP 法とほぼ同等の精度となっている.

著者らは, さらに教師なし学習法として Neural Gas 法についても同様に学習法を提案し, 数値実験を行った. 結果は, BP 法と同じであった.

5. まとめ

5.1 本研究の社会への貢献

SDG's の目的は, 持続可能社会の構築であり, 日本は, Society 5.0 を目指して超スマート社会の実現を通して貢献する. それでは, 本研究を含む安心・安全な機械学習は, 超スマート社会や持続可能な社会にどのように貢献するのであろうか. 持続可能な社会の構築において欠かせない要素としては, これまでに水や空気が知られており, 数百年に渡り安心・ 安全な環境作りが行われてきた. 一方, 20 世

表 4　提案された BP 法による数値実験の結果

		Iris	Wine	Sonar	BCW	Spam
A1	Learn(%)	2.41	0.51	1.81	2.38	6.34
	Test(%)	3.77	1.75	16.71	2.93	7.02
	LT	48799	24468	49211	25448	50000
A2	Learn(%)	1.73	0.50	1.19	2.34	6.18
	Test(%)	3.23	2.28	17.55	3.01	7.24
	LT	15371	1681	5150.08	920	4007
A3	Learn(%)	1.71	0.53	1.28	2.31	5.86
	Test(%)	2.83	2.06	15.74	2.91	6.70
	LT	28108	5004	12722	2884	2569
B1	Learn(%)	2.45	0.53	1.77	2.37	8.19
	Test(%)	3.20	2.08	16.43	2.89	8.59
	LT	48462	24411	49285	26006	50000
B2	Learn(%)	1.66	0.54	1.19	2.30	6.18
	Test(%)	3.00	1.92	16.05	2.91	7.29
	LT	15474	1682	4972	1313	3986
B3	Learn(%)	1.68	0.51	1.17	2.31	5.81
	Test(%)	3.07	2.14	16.52	2.86	6.64
	LT	27798	5116	12664	3413	4585
C1	Learn(%)	4.02	1.19	3.86	2.30	6.41
	Test(%)	4.87	3.58	18.83	2.99	7.19
	LT	49474	16333	43298	20489	50000
C2	Learn(%)	2.73	1.23	1.95	2.23	6.14
	Test(%)	5.33	3.97	18.14	3.01	6.91
	LT	6513	1057	7295	576	901
C3	Learn(%)	1.74	1.12	2.31	2.22	5.96
	Test(%)	4.03	3.97	18.38	3.02	6.71
	LT	13051	1919	20986	1621	1444

紀以降, コン ピュータや通信の発達に伴って, 個人が国や世界と直接関わりを持って生きていくことになってきている. 過去には人と人との関わりは, 直接または郵便等を通して情報交換を行うことが前提であった. 今日では, 人の関わりに情報が国境を越えて瞬時に到達する時代となっている. この時代になると個人の情報は簡単に収集できてビッグデータとなり, これらを使った新しい知識や情報がAI による解析を通して次々に生成されて, 個人や社会に還元される. また, ビッグデータは, 持続可能社会を支えるために, 次の再利用に向けて安全に保存される. これがいわゆる超スマート社会の望まれる形である.

それではこの社会でのビッグデータの処理サイクルが持続可能であるためには, 何が必要であろうか? そのためには, このサイクルが安心・安全に利活用できるシステムの環境作りが必要となる. 本研究を含むプライバシー保護に関する機械学習の研究は, この分野のインフラの構築に貢献するものであり, ユーザは, 個々の情報を提供して, 必要とする知識や情報を得るサイクルにおいて, 個人の特定や情報の漏洩についての不安から解放される.

例えば, 昨今問題となっている人流データは, 人がいつどこに何人いるかを把握できるデータである. スマホの位置情報やカメラ映像によって容易にビッグデータを収集できる. この情報は, 一般的にはマーケティング, 観光, 行政サービスや防災などへの利用が期待されている. COVID-19 に対しては, このデータに統計処理や AI 処理を施すことによって, 感染者のデータ分布, クラスタの存在や感染者の広がり方を可視化することができる. この結果を通して, 我々がいかに安心・安全に行動すべきかの知識や指針を知ることができる. ただし, この情報はプライバシー性が極めて高いので, 個人が特定できないように数値化する等, 慎重な取り扱いが必要となる. この問題に対して, 従来法と提案法は, ユーザに対してどのように安心・安全を与えるかをまとめると以下のようになる.

1) 暗号化を用いる方法では, 厳密にユーザの特定を許さない安心・安全な方法であるが, ビッグデータの増大に伴うAI 処理の対応の難しさや, 応答の遅れが強まると考えられる. すなわち, 刻々と変化する現地（被災地）等から送られるビッグデータをリアルタイムかつ正確に AI 処理することは難しい.

2) 連合学習は, ビッグデータの増大に伴う AI 処理には柔軟に対応できるが, ビッグデータを分散した各サーバでの情報の漏洩や散逸の問題は, これまでのクラウドシステムと同様に懸念される. すなわち, ユーザのデータに対するリスク対策が各サーバにおいて別途必要になる.

3) 提案法は, ビッグデータの増大に伴なう処理能力は, 連合学習とほぼ同等である. また, 個人データはあらかじめ分解された断片となり, これを用いた機械学習を実行することにより, 個人情報の漏洩や散逸の可能性を低く抑えることができる. すなわち, 提案法を用いた超スマート社会では, ユーザ情報は, 社会に分解保存という形で安全・安心に取り込まれて, 高度 AI 処理によって安全に処理される. ユーザは, 必要に応じて周辺情報や個人の安全な行動パターンを知ることができる. また, 提供された情報は, 次の再利用に向けて個人が特定されない形で安心・安全に保存される.

同様に, 病気, 防災や減災, 教育, 工業, 農業や株式等, 様々な分野でプライバシーを保護する AI 処理の応用が広がっている. それゆえ, これからのビッグデータの高度 AI 処理については, これを活用したときの価値と, プライバシーに対するリスクのバランスを超スマート社会の共通認識として利用していくことが必要となる. また, これらの分野における持続可能なサイクルが, 安定して継続されるインフラとして提供される時代に向けて, 多方面からのさらなる検討が望まれる. 本研究を含むプライバシー保護に関する機械学習の研究は, この分野のインフラの構築に貢献するものである.

5.2　提案法の位置づけと今後の展開

持続可能な社会を経済面から支える, 超スマート社会の構築に関して, プライバシーを保護する AI 処理の実現に関して, 本論文では, データを秘匿したまま機械学習を行

う方法として BP 法への適用を通してその有効性を示した. ビッグデータの AI 処理や視覚化においては, 結果に対する期待とデータ提供に対する不安が交錯する. この不安を取り除くために, 従来法としては, データを暗号化し機械学習を行い厳密にプライバシーを保護する方法と, データを複数の部分集合に分けて, それぞれの学習の結果を統合することにより学習を容易に実行できる連合学習が知られていた. 提案法では, あらかじめ各データを複数のランダムな断片に分解し, それぞ れを複数のサーバに記憶する. 学習は, 中央サーバと複数サーバとの分散処理により実現する.

提案法の優れた点は, 暗号化法と同様に学習時にオリジナルのデータを分解して使うことによりプライバシー保護に対するリスクを減らせることと, 学習に対する利活用については, 連合学習と同じように多くの問題に適用可能である. まとめると, 以下のようになる.

1) 持続可能な社会の実現に向けて, ビッグデータに対する高度 AI 処理に対する必要性が多くの分野で高まっている. これまでに, AI 処理の利活用による価値に重きを置いた研究や, セキュリティのリスクを重視した研究が行われてきた.

2) 本論文の提案法は, AI 処理に関して利活用に対する価値と, セキュリティのリスクに対するバランスが高度にとれた学習法である.

今後の展開としては, 超スマート社会における安心・安全な AI 処理のインフラ構築を目指して, 利活用とリスクの両面からより高い処理能力を持つ学習法の開発を目指したい. また, 技術的には, 強化学習への提案法の導入や連合学習の各サーバ上への提案法の導入による, セキュリティ向上が考えられる.

謝辞 本研究は公益財団法人 電気通信普及財団の助成を受けたものである.

参考文献

[1] United Nations Foundation: SUSTAINABLE DEVELOPMENT GOALS(online), 入手先 ⟨https://unfoundation.org/⟩ (2021.11.09)
[2] UNDP Ukraine: Transforming our world: the 2030 Agenda for Sustainable Development(online), 入手先 ⟨https://www.ua.undp.org/⟩ (2021.11.09)
[3] 内閣府: Society 5.0(online), 入手先 ⟨https://www.cao.go.jp/⟩ (2021.11.09)
[4] United Nations: Global Sustainable Development Report, 2015 edition(online), 入手先 ⟨https://www.un.org/en/desa⟩ (2021.11.09)
[5] Aggarwal C. C., and Yu P. S.: *Privacy-Preserving Data Mining: Models and Algorithms*, ISBN 978-0-387-70991-8, Springer-Verlag (2009)
[6] Shamir A.: *How to share a secret*, Comm. ACM, Vol. 22, No. 11, pp. 612-613 (1979)
[7] Beimel A.: *Secret-sharing schemes: a survey*, Proc. of the Third international conference on Coding and cryptology (IWCC 11) (2011)
[8] Canetti R., Feige U., Goldreich O., and Naor M.: *Adaptively secure multi-party computation*, STOC' 96, pp. 639-648 (1996)
[9] Cramer R., Damgard I., and Maurer U.: *General secure multi-party computation from any linear secret-sharing scheme*, EUROCRYPT', pp.331-339 (2000)
[10] Ben-David A., Nisan N., and Pinkas B.: *Fair play MP: a system for secure multi-party computation*, ACM CCS' 08 (2008)
[11] Gentry C.: *Fully Homomorphic Encryption Using Ideal Lattices*, STOC2009, pp.169-178 (2009)
[12] HElib: An Implementation of homomorphic encryption(online), 入手先 ⟨https://github.com/shaih/HElib⟩ (2021/11/09)
[13] Yang Q., Li Y., Chen T., and Tong Y.: *Federated Machine Learaning : Concept and Applications*, ACM Trans. Intell. Syst. Technol., Vol.10, No.2, Article 12 (2019)
[14] Konen J., McMahan H. B., Yu F. X., Richtrik P., Suresh A. T., Bacon D.: *Federated Learning: Strategies for Improving Communication Efficiency*, arXiv:1610.05492 (2017)
[15] Rathna S. S., Karthikeyan T.: *Survey on Recent Algorithms for Privacy Preserving Data mining*, International Journal of Computer Science and Information Technologies, Vol. 6 (2), pp. 1835-1840 (2015)
[16] Yuan J., Yu S.: *Privacy Preserving Back-Propagation Neural Network Learning Made Practical with Cloud Computing*, IEEE Trans. On Parallel and Distributed Systems, Vo.25, Issue 1, pp.212-221 (2013)
[17] Schlitter N.: *A Protocol for Privacy Preserving Neural Network Learning on Horizontal Partitioned Data*, Privacy Statistics in Databases (PSD) (2008)
[18] Miyajima H., Miyajima H. and Shiratori N.: *Fast and Secure Back-Propagation Learning using Vertically Partitioned Data with IoT*, CANDAR 2019 : The Seventh International Symposium on Computing and Networking, Nagasaki, November, pp.450-454 (2019)
[19] Miyanishi Y., Kanaoka A., Sato F., Han X., Kitagami S., Urano Y., Shiratori N.: *New Methods to Ensure Security to Increase User's Sense of Safety in Cloud Services*, Proc. of The 14th IEEE Int. Conference on Scalable Computing and Communications (ScalCom-2014), pp.859-865 (2014)
[20] Miyajima H., Shigei N., Miyajima H., Miyanishi Y., Kitagami S, and Shiratori N.: *New Privacy Preserving Back Propagation Learning for Secure Multiparty Computation*, IAENG International Journal of Computer Science, vol.43, no.3, pp.270-276 (2016)
[21] Ruder S.: *An Overview of Gradient Descent Optimization Algorithms*, 入手先 ⟨http://ruder.io/optimizing-gradient-descent/⟩(2016) (2018.05.14).
[22] Gupta M. M., Jin L., Honma N.: *Static and Dynamic Neural Networks*, IEEE Pres, Wiley-Interscience (2003).
[23] Miyajima H., Miyajima H. and Shiratori N.: *Proposal of Fast and Secure Clustering Method for IoT*, International MultiConference of Engineers and Computer Scientists 2019 (2019).
[24] UCI Repository of Machine Learning Databases and Domain Theories: datasets(online), 入手先 ⟨https://archive.ics.uci.edu/ml/datasets.php⟩ (2021.11.09).

ⓒ 2021 Information Processing Society of Japan