(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2018-45679
(P2018-45679A)
(43)公開日　平成30年3月22日(2018.3.22)

(51)Int.Cl.			F I			テーマコード（参考）
G06N	3/08	(2006.01)	G06N	3/08		5J104
H04L	9/14	(2006.01)	H04L	9/00	641	

審査請求 未請求　請求項の数 13　ＯＬ　（全 34 頁）

(21)出願番号　特願2017-120278 (P2017-120278)
(22)出願日　平成29年6月20日 (2017.6.20)
(31)優先権主張番号　特願2016-175700 (P2016-175700)
(32)優先日　平成28年9月8日 (2016.9.8)
(33)優先権主張国　日本国 (JP)

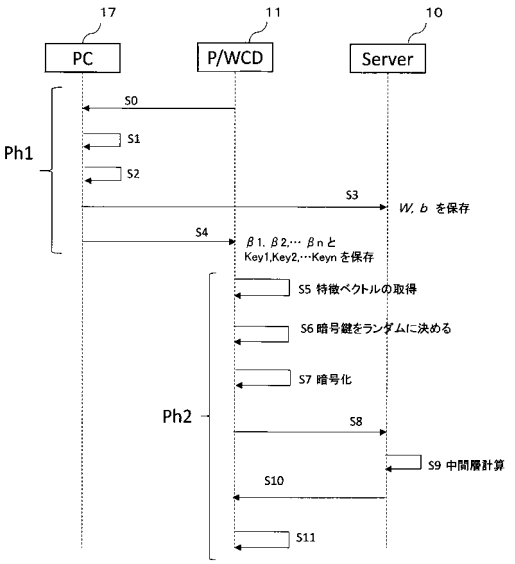
（特許庁注：以下のものは登録商標）
１．ＡＮＤＲＯＩＤ

(71)出願人　506301140
公立大学法人会津大学
福島県会津若松市一箕町大字鶴賀字上居合９０番地
(74)代理人　100094525
弁理士　土井　健二
(74)代理人　100094514
弁理士　林　恒徳
(72)発明者　趙　強福
福島県会津若松市一箕町大字鶴賀字上居合９０番地　公立大学法人会津大学内
(72)発明者　橋本　雅人
福島県会津若松市一箕町大字鶴賀字上居合９０番地　公立大学法人会津大学内

Ｆターム(参考)　5J104 AA16　EA04　EA18　NA02　NA37
PA02

(54)【発明の名称】携帯端末を用いた察知エージェントシステム、察知エージェントシステムにおける機械学習方法、及びこれを実施するためのプログラム

(57)【要約】　（修正有）
【課題】機械学習による大規模な計算処理の不都合を解決し、ユーザの様々な情報を察知して、その情報を基にユーザに役立つ情報の提供や、問題解決する携帯端末を用いた察知エージェントシステムを提供する。
【解決手段】携帯端末を用いた察知エージェントシステムであって、携帯端末と、携帯端末に接続するサーバを有する。携帯端末は、ユーザから取得する情報に含まれる特徴ベクトルを暗号化し、次いで、暗号化された特徴ベクトルをニューラルネットワークの入力層としてサーバに送信する。サーバは、前記暗号化された特徴ベクトルを受信して、前記ニューラルネットワークの入力層から隠れ層を計算し、前記隠れ層の計算結果を前記携帯端末に送信する。携帯端末は更に、前記サーバからの隠れ層の計算結果から出力層の計算を行う。
【選択図】図４



【特許請求の範囲】
【請求項１】
携帯端末を用いた察知エージェントシステムであって、
携帯端末と、
前記携帯端末に接続されるサーバを有し、
前記携帯端末は、
ユーザから取得される情報から取得した特徴ベクトルを暗号化し、次いで、前記暗号化された特徴ベクトルをニューラルネットワークの入力層として前記サーバに送信し、
前記サーバは、
前記暗号化された特徴ベクトルを受信して、前記ニューラルネットワークの入力層から隠れ層を計算し、
前記隠れ層の計算結果を前記携帯端末に送信し、
前記携帯端末は、更に、前記サーバからの隠れ層の計算結果から出力層の計算を行う、
ことを特徴とする察知エージェントシステム。
【請求項２】
携帯端末を用いた察知エージェントシステムであって、
携帯端末と、
前記携帯端末に接続されるサーバを有し、
前記携帯端末は、
ユーザから取得される情報から取得した特徴ベクトルを第１の鍵により暗号化し、次いで、前記暗号化された特徴ベクトルをニューラルネットワークの入力層として前記サーバに送信し、
前記サーバは、
前記暗号化された特徴ベクトルを受信して、大きな乱数行列から部分重み行列を決定するための第２の鍵を用い、前記第２の鍵により決定される部分重み行列を用いて前記ニューラルネットワークの入力層から隠れ層を計算し、前記隠れ層の計算結果を前記携帯端末に送信し、
前記携帯端末は更に、前記隠れ層の計算結果の要素のうちその後の処理に用いない要素を決定するための第３の鍵を用いて、前記隠れ層の計算結果の要素を隠す処理を行い、その後、出力層の計算を行う、
ことを特徴とする察知エージェントシステム。
【請求項３】
請求項２において、
前記第１の鍵により、前記ユーザからのデータがベクトル置換によりを暗号化される、
ことを特徴とする察知エージェントシステム。
【請求項４】
請求項１乃至３において、
前記ニューラルネットワークは、機械学習アルゴリズムであって、
前記機械学習アルゴリズムは、処理フェーズとして、学習フェーズと、識別分類フェーズを有する、
ことを特徴とする察知エージェントシステム。
【請求項５】
請求項１において、
さらに、コンピュータを有し、
前記コンピュータは、学習フェーズとして、
前記携帯端末において前記暗号化に用いる複数の暗号鍵と、前記複数の暗号鍵に対応して前記出力層の計算に用いる複数の重みβを生成し、更に
前記サーバにおいて前記暗号化された特徴ベクトルに対して隠れ層の計算をするための重みＷとバイアスｂを生成する、
ことを特徴とする携帯端末を用いた察知エージェントシステム。

【請求項６】
請求項５において、
さらに、識別分類フェーズとして、
前記携帯端末は、前記複数の暗号鍵によりランダムに決定した暗号鍵を用いて、ユーザの入力した特徴ベクトルを暗号化し、
前記サーバは、前記暗号化された特徴ベクトルを、前記重みＷとバイアスｂを用いて隠れ層の計算をし、
前記携帯端末は、更に前記サーバにおける隠れ層の計算結果に対し、前記暗号化に用いた暗号鍵に対応する重みβを用いて出力層の計算を行う、
ことを特徴とする携帯端末を用いた察知エージェントシステム。
【請求項７】
請求項１乃至６の何れかにおいて、
前記サーバは、クラウドサーバであって、
前記携帯端末はインターネットを通して前記サーバに接続する、
ことを特徴とする携帯端末を用いた察知エージェントシステム。
【請求項８】
請求項２において、
さらに、コンピュータを有し、
前記コンピュータは、学習フェーズとして、
隠れニューロン数を決定し、
ユーザから取得される情報から取得した特徴ベクトルを暗号化するための第１の鍵と、大きな乱数行列から部分重み行列を決定するための第２の鍵と、隠れ層の計算結果の要素のうちその後の処理に用いない要素を決定するための第３の鍵を生成し、更に出力層の重みを算出する、
ことを特徴とする携帯端末を用いた察知エージェントシステム。
【請求項９】
請求項８において、
さらに、識別分類フェーズとして、
前記携帯端末は、
前記第１、第２、第３の鍵と前記出力層の重みを保存し、前記第１の鍵を用いて、前記特徴ベクトルを暗号化し、
前記暗号化された特徴ベクトルと、前記第２の鍵及び隠れニューロン数を前記サーバに送信し、
前記サーバは、
前記第２の鍵及び隠れニューロン数により隠れ層の重みを求め、隠れ層を計算して携帯端末に送信し、更に
前記携帯端末は、前記計算された隠れ層の出力から前記第３の鍵により後の計算に用いない要素を決定する、
ことを特徴とする携帯端末を用いた察知エージェントシステム。
【請求項１０】
携帯端末を用いた察知エージェントシステムにおける機械学習方法であって、
学習フェーズと識別分類フェーズを有し、
前記学習フェーズとして、コンピュータによる、
前記携帯端末における転置式暗号化のための異なるｎ個の暗号鍵（key_1－key_n）と、前記暗号鍵に対応して、ニューラルネットワークの出力層の計算のための重みβを生成し、更に、サーバにおける前記ニューラルネットワークの隠れ層の計算に用いるニューロンの重みＷとバイアスｂを生成するステップと、
前記識別分類フェーズとして、
前記携帯端末による前記ｎ個の暗号鍵をランダムに用いて、特徴ベクトルを暗号化するステップと、

前記サーバによる前記暗号化された特徴ベクトルを前記ニューロンの重みWとバイアスｂを用いて前記ニューラルネットワークの隠れ層の計算をするステップと、
前記携帯端末により更に、前記隠れ層の計算結果に対し、前記特徴ベクトルを暗号化に用いた暗号鍵に対応する重みβを用いて前記ニューラルネットワークの出力層の計算を行うステップを、
有することを特徴とする機械学習方法。
【請求項１１】
携帯端末を用いた察知エージェントシステムにおける機械学習方法であって、
学習フェーズと、識別分類フェーズを有し、
前記学習フェーズとして、
コンピュータによる、
隠れニューロン数を決定するステップと、
ユーザから取得される情報から取得した特徴ベクトルを暗号化するための第１の鍵と、大きな乱数行列から部分重み行列を決定するための第２の鍵と、隠れ層の計算結果の要素のうちその後の処理に用いない要素を決定するための第３の鍵を生成するステップと
出力層の重みを算出するステップを有し、
前記携帯端末に第１、第２、第３の鍵と前記出力層の重みを保存するステップを有し、
前記識別分類フェーズとして、
前記携帯端末による前記第１の鍵を用いて、前記特徴ベクトルを暗号化するステップと、
暗号化された特徴ベクトルと、前記第２の鍵及び隠れニューロン数をサーバに送信するステップと、
前記サーバにより、前記第２の鍵及び隠れニューロン数により隠れ層の重みを求め、隠れ層を計算して携帯端末に送信するステップと、
携帯端末で、前記計算された隠れ層の出力から前記第３の鍵により後の計算に用いない要素を決定し、更に出力層の計算を行うステップを有する、
ことを特徴とする機械学習方法。
【請求項１２】
携帯端末とサーバを有する察知エージェントシステムにおける機械学習方法を実行するプログラムであって、
コンピュータに、
前記携帯端末における転置式暗号化のための異なるｎ個の暗号鍵（key_1－key_n）と、前記暗号鍵に対応して、ニューラルネットワークの出力層の計算のための重みβを生成し、更に、サーバにおける前記ニューラルネットワークの隠れ層の計算に用いるニューロンの重みWとバイアスｂを生成するステップを実行させ、
前記携帯端末に、
前記ｎ個の暗号鍵をランダムに用いて、特徴ベクトルを暗号化するステップを実行させ、
前記サーバに
前記暗号化された特徴ベクトルを前記ニューロンの重みWとバイアスｂを用いて前記ニューラルネットワークの隠れ層の計算をするステップを実行させ、更に、
前記携帯端末に、
前記隠れ層の計算結果に対し、前記特徴ベクトルを暗号化に用いた暗号鍵に対応する重みβを用いて前記ニューラルネットワークの出力層の計算を行うステップを実行させる、
ことを特徴とするプログラム。
【請求項１３】
携帯端末を用いた察知エージェントシステムにおける機械学習方法を実行するプログラムであって、
学習フェーズと、識別分類フェーズを有し、
前記学習フェーズとして、

コンピュータに、
隠れニューロン数を決定させるステップと、
ユーザから取得される情報から取得した特徴ベクトルを暗号化するための第１の鍵と、大きな乱数行列から部分重み行列を決定するための第２の鍵と、隠れ層の計算結果のうちその後の処理に用いない要素を決定するための第３の鍵を生成するステップと
出力層の重みを算出するステップを実行させ、
前記携帯端末に前記第１、第２、第３の鍵と前記出力層の重みを保存させるステップを実行させ、更に、
前記識別分類フェーズとして、
前記携帯端末に、
前記第１の鍵を用いて、前記特徴ベクトルを暗号化させるステップと、
暗号化された特徴ベクトルと、前記第２の鍵及び隠れニューロン数をサーバに送信させるステップを実行させ、
前記サーバに、
前記第２の鍵及び隠れニューロン数により隠れ層の重みを求め、隠れ層を計算して携帯端末に送信させるステップを実行させ、更に
前記携帯端末に、
前記隠れ層の計算結果のうちその後の処理に用いない要素を前記第３の鍵を用いて決定させ、その後、出力層を計算させる、
ことを特徴とするプログラム。

【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、察知エージェントシステムに関し、特に携帯して利用可能な情報処理機能を有する通信機器や情報機器（以降これらを携帯端末と称す）を用いた察知エージェントシステムに関する。さらに、察知エージェントシステムにおける機械学習方法、及びこれを実施するためのプログラムに関する。
【背景技術】
【０００２】
察知エージェントとは、ユーザの様々な情報を察知して、その情報を基にユーザに役立つ情報の提供や、問題解決を行うことを目的としたさまざまな分野に応用ができるシステムである。
【０００３】
かかる察知エージェントにＥＬＭ（Extreme Learning Machine）に基づく機械学習手法を取り入れることによりユーザにカスタマイズされたエージェントを提供することが可能である。
【０００４】
ここで、脳の中には多数の神経細胞（ニューロン）があり、ニューロンは、他のニューロンからの信号を受け、入力される信号の状態から所定の信号を他の多数のニューロンに受け渡しして情報処理を行っている。このような脳における仕組みをコンピュータにより実現するものが、ニューラルネットワークである。
【０００５】
ＥＬＭは、一種のニューラルネットワークであり、大量の既知のデータから情報を抽出し、その情報をもとに未知のデータが何であるかを推論、分類する技術である。ここで、以降の説明では、「情報を抽出」することを学習、「推論、分類」することを分類と表示する。
【０００６】
図１は、一般的なＥＬＭニューラルネットワークを示す図である。図において、ニューラルネットワーク１は、入力層１ａ、隠れ層（中間層）１ｂ、出力層１ｃを有する。
【０００７】

隠れ層１ｂの重みＷとバイアスｂは、乱数である。βは出力層１ｃの重みである。これら重みと係数は、事前のトレーニングにより決められる。

【０００８】

ここで、入力の特徴ベクトルをｘとすると、特徴ベクトルｘの関数は、次式で表される。

【０００９】

【数１】

$$f(x) = \mathrm{sign}\left(h(x) \cdot \beta \right) \quad \cdots (1)$$

【００１０】

【数２】

$$\mathrm{h}(x) = G(W \cdot x + b) \quad \cdots (2)$$

【００１１】

【数３】

$$G(z) = \begin{pmatrix} g(z_1) \\ \vdots \\ g(z_{N_h}) \end{pmatrix} \quad \cdots (3)$$

【００１２】

【数４】

$$W = \begin{pmatrix} w_{11} & \cdots & w_{1N_f} \\ \vdots & \ddots & \vdots \\ w_{N_h 1} & \cdots & w_{N_h N_f} \end{pmatrix} \quad \mathrm{b} = \begin{pmatrix} b_1 \\ \vdots \\ b_{N_h} \end{pmatrix} \quad \cdots (4)$$

【００１３】

活性化関数としてシグモイド関数g(z)を用いる。以下はシグモイド関数の一例である。

【００１４】

【数５】

$$g(z) = \frac{1}{1+exp(-\lambda \cdot z)} \quad \cdots (5)$$

【００１５】

上記式（１）－（４）において、N_hは隠れニューロンの数、Ｗは入力層から隠れ層の間の重み、ｂはバイアスである。βは隠れ層から出力層の間の重みである。また、N_fは特徴ベクトルｘの次元数である。

【００１６】

基本的なＥＬＭの学習方法を示すと次の手順のようである。

【００１７】

学習データが、

【００１８】

【数６】

$$\Omega = \left\{ \left(x_i, t_j \right) | x_j \in R^{N_f}, t_j \in \{-1,1\}, j = 1, \ldots, N_d \right\}$$

【００１９】

である。（但し、N_fは特徴ベクトルの次元数、N_cはクラス数、N_dはデータ数）

１）まず、Ｗとｂを乱数で初期化する。通常、乱数の範囲は、[-1,1]である。行列Ｗのサイズは、$N_h \times N_f$，ｂはN_h次元のベクトルである。

２）隠れ層の出力の行列を計算する。
【００２０】
【数７】

$$\mathrm{H}=\begin{pmatrix} h(x_1)^t \\ \vdots \\ h(x_{N_d})^t \end{pmatrix}=\begin{pmatrix} G(W\cdot x+b)^t \\ \vdots \\ G(W\cdot x_{N_d}+b)^t \end{pmatrix} \quad \cdots(6)$$

【００２１】
次いで、
３）出力層の重みベクトルβを求める。
【００２２】
【数８】

$$T=H\cdot\beta$$

$$\beta=H^{+}\cdot T \quad \cdots(7)$$

【００２３】
このとき、H^{+}は式（６）の疑似逆行列である。Ｔは次に示す行列式である。
【００２４】
【数９】

$$T=\begin{pmatrix} t_1 \\ \vdots \\ t_{N_d} \end{pmatrix} \quad \cdots(8)$$

【００２５】
この方法によって求められたＷとβと式（１）を用いて、分類処理を行う。
【００２６】
一方、近年、スマートフォンをはじめとした携帯端末（Portable/Wearable Computing Device: P/WCD）が急速に普及し、今後も利用者が増えることが予想される。様々なアプリケーションがかかる携帯端末に向けて開発され、電話やメールに限らず多くの場面で利用されている。
【００２７】
したがって、機械学習の手法を取り入れたアプリケーションの一つとして察知エージェントを携帯端末で実行できる場合、ユーザの様々な情報を察知して、その情報を基にユーザに役立つ情報の提供や、問題解決を行うことが可能である。
【先行技術文献】
【特許文献】
【００２８】
【特許文献１】特開平８－２０６０８８号公報
【特許文献２】特許第５９１６４６６号公報
【特許文献３】特開２０１４－２２９１２４号公報
【非特許文献】
【００２９】
【非特許文献１】M. Lichman. UCI Machine Learning repository, 2013. URL:http://archive.ics.uci.edu/ml.
【非特許文献２】mldata.org.URL:http://mldata.org/.
【非特許文献３】Stefan Van Der Walt, S Chris Colbert, and Gael Varoquaux. “The NumPy array: a structure for efficient numerical computation”. In: Computing in Science & Engineering 13.2 (2011), pp. 22-30.

【非特許文献４】Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python. [Online; accessed 2016-04-05]. 2001-. URL:http://www.scipy.org.
【非特許文献５】F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: Journal of Machine Learning Research 12 (2011), pp. 2825-2830.
【非特許文献６】Lars Buitinck et al. “API design for machine learning software: experiences from the scikit-learn project”. In:ECML PKDD Workshop: Languages for Data Mining and Machine Learning. 2013, pp. 108-122.
【非特許文献７】Marcel Hellkamp. bottle. 2014. URL:http://bottlepy.org/.
【非特許文献８】Inc Square. OkHttp. 2014. URL: http://square.github.io/okhttp/.
【非特許文献９】Jackson. URL: http://wiki.fasterxml.com/JacksonHome.
【非特許文献１０】Florian Tramer et al. "Stealing machine learning models via prediction apis". In: USENIX Security. 2016.
【非特許文献１１】Masato Hashimoto, Yuya Kaneda, and Qiangfu Zhao. “An ELM-based privacy preserving protocol for cloud systems”. In IEEE Symposium Series on Computational Intelligence (2016).
【非特許文献１２】Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. “Extreme learning machine: theory and applications”. In: Neurocomputing 70.1 (2006), pp. 489-501.
【非特許文献１３】Guang-Bin Huang et al. “Extreme learning machine for regression and multiclass classification”. In: Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 42.2 (2012), pp. 513-529.
【非特許文献１４】nginx. URL:https://nginx.org.
【非特許文献１５】uWSGI. URL:https://uwsgi-docs.readthedocs.io/en/latest/.
【非特許文献１６】ReactiveX. URL:http://reactivex.io/.
【非特許文献１７】Retrofit. URL:https://square.github.io/retrofit/.
【非特許文献１８】gson. URL:https://github.com/google/gson.
【非特許文献１９】M.Lichman.UCI Machine Learning Repository. http://archive.ics.uci.edu/ml. 2013
【非特許文献２０】australian. URL: https://archive.ics.uci.edu/ml/datasets/Statlog+(Australian+Credit+Approval).
【非特許文献２１】QSAR. URL: https://archive.ics.uci.edu/ml/datasets/QSAR+biodegradation.
【非特許文献２２】gisette. URL:https://archive.ics.uci.edu/ml/datasets/Gisette.
【非特許文献２３】satimage. URL:https://arachive.ics.uci.edu/ml/datasets/Statlog+(Landsat+Satellite)

【発明の概要】
【発明が解決しようとする課題】
【００３０】
上記したように、察知エージェントを携帯端末に実装することは、より適切なサービスをより多くのユーザに提供が可能であるという有利さがある。しかし、携帯端末における計算資源は、一般のコンピュータよりも劣っている。さらにバッテリ駆動であるために、携帯端末において察知エージェントのための機械学習による大規模な計算処理を行うことが困難である。
【００３１】
このような性能と計算コストのトレードオフの問題を解決する方法として、クラウドを用いること、即ち携帯端末上での機械学習処理が困難である場合、クラウドサーバに行わせることが想定される。
【００３２】
この方法は、ネットワークの高速化、安定化に伴い、多くの携帯端末用のアプリケーシ

ョンの設計方法にもなっている。携帯端末ではデータの入出力のみを行い、計算処理はクラウドサーバで行う方法である。

【００３３】

しかし、クラウドシステムは既知の問題として、情報漏えいの問題と、それによるプライバシー侵害の問題がある。全ての察知エージェントの処理をクラウドで行う場合、クラウド側でユーザのプライバシー情報の解析が可能になる。また、クラウドで情報漏えいが生じた場合は、第三者にユーザの個人情報が流出してしまうという恐れがある。

【００３４】

したがって、本発明の目的は、かかる機械学習による大規模な計算処理の不都合を解決し、ユーザの様々な情報を察知して、その情報を基にユーザに役立つ情報の提供や、問題解決する携帯端末を用いた察知エージェントシステムを提供することにある。

【課題を解決するための手段】

【００３５】

上記課題を解決する本発明に従う察知エージェントの第一の側面は、携帯端末を用いた察知エージェントシステムであって、携帯端末と、前記携帯端末に接続されるサーバを有し、前記携帯端末は、ユーザから取得される情報から取得した特徴ベクトルを暗号化し、次いで、前記暗号化された特徴ベクトルをニューラルネットワークの入力層として前記サーバに送信し、前記サーバは、前記暗号化された特徴ベクトルを受信して、前記ニューラルネットワークの入力層から隠れ層を計算し、前記隠れ層の計算結果を前記携帯端末に送信し、前記携帯端末は、更に、前記サーバからの隠れ層の計算結果から出力層の計算を行うことを特徴とする。

【００３６】

上記課題を解決する本発明に従う察知エージェントの第二の側面は、携帯端末を用いた察知エージェントシステムであって、携帯端末と、前記携帯端末に接続されるサーバを有し、前記携帯端末は、ユーザから取得される情報から取得した特徴ベクトルを第１の鍵により暗号化し、次いで、前記暗号化された特徴ベクトルをニューラルネットワークの入力層として前記サーバに送信し、前記サーバは、前記暗号化された特徴ベクトルを受信して、大きな乱数行列から部分重み行列を決定するための第２の鍵を用い、前記第２の鍵により決定される部分重み行列を用いて前記ニューラルネットワークの入力層から隠れ層を計算し、前記隠れ層の計算結果を前記携帯端末に送信し、前記携帯端末は更に、前記隠れ層の計算結果の要素のうちその後の処理に用いない要素を決定するための第３の鍵を用いて、前記隠れ層の計算結果の要素を隠す処理を行い、その後、出力層の計算を行うことを特徴とする。

【００３７】

上記課題を解決する本発明に従う察知エージェントの第二の側面において、前記第１の鍵により、前記ユーザからのデータがベクトル置換によりを暗号化されることを特徴とする。

【００３８】

上記課題を解決する本発明に従う察知エージェントは、前記第一、第二の側面において、前記ニューラルネットワークは、機械学習アルゴリズムであって、前記機械学習アルゴリズムは、処理フェーズとして、学習フェーズと、識別分類フェーズを有することを特徴とする。

【００３９】

上記課題を解決する本発明に従う察知エージェントの第一の側面において、さらに、コンピュータを有し、前記コンピュータは、学習フェーズとして、前記携帯端末において前記暗号化に用いる複数の暗号鍵と、前記複数の暗号鍵に対応して前記出力層の計算に用いる複数の重みβを生成し、更に前記サーバにおいて前記暗号化された特徴ベクトルに対して隠れ層の計算をするための重みＷとバイアスｂを生成することを特徴とする。

【００４０】

上記課題を解決する本発明に従う察知エージェントの第一の側面において、前記識別分

類フェーズとして、前記携帯端末は、前記複数の暗号鍵によりランダムに決定した暗号鍵を用いて、ユーザの入力した特徴ベクトルを暗号化し、前記サーバは、前記暗号化された特徴ベクトルを、前記重みＷとバイアスｂを用いて隠れ層の計算をし、前記携帯端末は、更に前記サーバにおける隠れ層の計算結果に対し、前記暗号化に用いた暗号鍵に対応する重みβを用いて出力層の計算を行うことを特徴とする。

【００４１】

上記課題を解決する本発明に従う察知エージェントは、前記第一、第二の側面において、前記サーバは、クラウドサーバであって、前記携帯端末はインターネットを通して前記サーバに接続することを特徴とする。

【００４２】

上記課題を解決する本発明に従う察知エージェントの第二の側面において、さらに、コンピュータを有し、前記コンピュータは、学習フェーズとして、隠れニューロン数を決定し、ユーザから取得される情報から取得した特徴ベクトルを暗号化するための第１の鍵と、大きな乱数行列から部分重み行列を決定するための第２の鍵と、隠れ層の計算結果の要素のうちその後の処理に用いない要素を決定するための第３の鍵を生成し、更に出力層の重みを算出することを特徴とする。

【００４３】

上記課題を解決する本発明に従う察知エージェントの第二の側面において、さらに、前記識別分類フェーズとして、前記携帯端末は、前記第１、第２、第３の鍵と前記出力層の重みを保存し、前記第１の鍵を用いて、前記特徴ベクトルを暗号化し、前記暗号化された特徴ベクトルと、前記第２の鍵及び隠れニューロン数を前記サーバに送信し、前記サーバは、前記第２の鍵及び隠れニューロン数により隠れ層の重みを求め、隠れ層を計算して携帯端末に送信し、更に前記携帯端末は、前記計算された隠れ層の出力から前記第３の鍵により後の計算に用いない要素を決定、更に出力層を計算することを特徴とする。

【００４４】

上記した本発明に従うプロトコルにおいて、携帯端末からサーバに送信されるデータは暗号化されたものである。さらに隠れ層のニューロンの重み行列は、完全に乱数である。従って、サーバでの計算、及び計算結果は優位な情報を持たず、よってユーザの情報は守られる。

【図面の簡単な説明】

【００４５】

【図１】一般的なニューラルネットワークを示す図である。

【図２】本発明に従う携帯端末を用いた察知エージェントシステムの概念図である。

【図３】図２に対応するサーバ１０と、携帯端末１１の構成例ブロック図である。

【図４】本発明に従う察知エージェントシステムの処理フローである。

【図５】隠れ層が複数であっても、サーバ１０での演算が可能であることを示す図である。

【図６】Australianデータベースを使用したときの分類時間の評価実験による結果を示すグラフである。

【図７】Satimageデータベースを使用したときの分類時間の評価実験による結果を示すグラフである。

【図８】USPSデータベースを使用したときの分類時間の評価実験による結果を示すグラフである。

【図９】Colon cancerデータベースを使用したときの分類時間の評価実験による結果を示すグラフである。

【図１０】Dexterデータベースを使用したときの分類時間の評価実験による結果を示すグラフである。

【図１１】Farm-adsデータベースを使用したときの分類時間の評価実験による結果を示すグラフである。

【図１２】第２の実施例に対応するニューラルネットワークを示す図である。

【図１３】第２の実施例に対応する学習(トレーニング)の処理フロー図である。
【図１４】第２の実施例に従う分類処理フロー図である。
【図１５Ａ】Australianデータセットにおける１０回の５分割交差検定による正答率を示す図である。
【図１５Ｂ】ＱＳＡＲデータセットにおける１０回の５分割交差検定による正答率を示す図である。
【図１５Ｃ】Gisetteデータセットにおける１０回の５分割交差検定による正答率を示す図である。
【図１５Ｄ】Satimageデータセットにおける１０回の５分割交差検定による正答率を示す図である。
【図１６Ａ】Nexus6におけるAustralianデータセットでの分類処理時間を示す図である。
【図１６Ｂ】Nexus6におけるＱＳＡＲデータセットでの分類処理時間を示す図である。
【図１６Ｃ】Nexus6におけるGisetteデータセットでの分類処理時間を示す図である。
【図１６Ｄ】Nexus6におけるSatimageデータセットでの分類処理時間を示す図である。
【図１７Ａ】Moto Z payにおけるAustralianデータセットでの分類処理時間を示す図である。
【図１７Ｂ】Moto Z payにおけるＱＳＡＲデータセットでの分類処理時間を示す図である。
【図１７Ｃ】Moto Z payにおけるGisetteデータセットでの分類処理時間を示す図である。
【図１７Ｄ】Moto Z payにおけるSatimageデータセットでの分類処理時間を示す図である。
【発明を実施するための形態】
【００４６】
以下に、本発明の実施例を添付の図面に従い説明する。この実施例は本発明の理解を容易とするためのものであり、本発明の適用は、これら実施例に限定されるものではない。また、本発明の保護の範囲は、特許請求の範囲と同一又は類似の範囲に及ぶ。
【００４７】
[第１の実施例]
図２は、本発明に従う携帯端末を用いた察知エージェントシステムの第１の実施例の概念図である。
【００４８】
機械学習のためのニューラルネットワークは、入力層１ａ、中間層（隠れ層）１ｂ、出力層１ｃを有する。本発明の特徴は、ニューラルネットワークの一部即ち、中間層１ｂの演算をサーバ１０に行わすことにある。
【００４９】
ここでサーバ１０として、好ましい形態としてクラウドサーバを使用するが、独立した固有のコンピュータサーバであってもよい。
【００５０】
事前に携帯端末１１には、出力層１ｃの演算のための出力ニューロンの重みβ_kが与えられている。一方、サーバ１０には、中間層１ｂの演算のための隠れニューロンの重みＷとバイアスｂが与えられている。
【００５１】
携帯端末１１においては、ユーザの入力に対する特徴ベクトルｘを暗号化する。そして、暗号化された特徴ベクトルｘを、サーバ１０に送る。
【００５２】
サーバ１０は、重みＷを用いて上記式（２）の計算をして中間層１ｂの演算をする。ついで、携帯端末１１は、重みβ_kを用いて上記式（１）の計算を行い、最終結果を得る。
【００５３】
かかる場合、サーバ１０での処理は、式（２）のみになる。ＥＬＭにおいてＷは乱数で

あるため、意味のある情報ではない。そのため、サーバ１０上のモデルの保護は保証される。また、サーバ１０で、全ての分類を行わないため、第三者にサーバ１０のデータを見られてもユーザの計算の意図を把握することは困難である。

【００５４】

さらに、サーバ１０においても、ユーザの計算目的や計算に使用するデータが何であるかを判断することが出来ない。したがって、これにより携帯端末１１を使用するユーザのプライバシーの保護が可能である。

【００５５】

図３は、図２に対応するサーバ１０と、携帯端末１１の構成例ブロック図である。サーバ１０と携帯端末１１は、ＷｉＦｉあるいは、ブルーツース等の無線通信機能１２を介して接続される。また必要であれば、携帯端末１１が有線接続によりサーバ１０に接続されてもよい。

【００５６】

携帯端末１１において、特徴ベクトルｘを暗号化部２０により暗号化してサーバ１０側に送信する。この暗号化部２０における暗号化のために、複数の暗号化キー（key_1～key_n）とそれに対する重みβ$_1$～β$_n$の組を暗号鍵と重みβの対応テーブル１３に事前に用意する。そして、分類毎に、これら暗号化キーと重みの組をランダムに切り替えて、暗号化部２０で使用することで安全性が高められる。

【００５７】

サーバ１０での処理は、入力される暗号化された特徴ベクトルｘに対して、演算部１４により中間層（隠れ層）１ｂの演算を、あらかじめ通知されている重みＷとバイアスｂの乱数を用いて行列計算を行う。

【００５８】

したがって、サーバ１０における処理において、データが第三者に知られてもセキュリティ上の不都合はない。すなわち、サーバ１０側では、ユーザの計算の意図、計算に使うデータがどのような意味を持つものであるかを判断することができない。

【００５９】

サーバ１０における演算の結果が、携帯端末１１に送られると、携帯端末１１の演算部１５で出力層１ｃの演算を行いクラスラベル１６が得られる。この際、携帯端末１１ではサーバ１０に特徴ベクトルを暗号化する際に使用した鍵(key_i)に対応する重みβを用いて出力層１ｃの演算を行う。

【００６０】

図４は、本発明に従う察知エージェントシステムの処理フローである。処理は、学習フェーズPh1と分類フェーズPh2で構成される。

【００６１】

「学習フェーズPh1」

学習フェーズPh1では、ニューラルネットワークで使用する先に図１に関して説明した隠れ層１ｂの重みＷとバイアスｂ、更に出力層１ｃの重みβを求める処理を行う。かかる処理は、コンピュータ１７として例えば、パーソナルコンピュータを用いて行うことも可能である。

【００６２】

コンピュータ１７において、携帯端末１１からユーザのさまざまな情報データを収集してデータベースを作成する（ステップＳ０）。ここで、ユーザの情報データは、察知エージェントをどのように使用するかによってその内容が変わる。

【００６３】

ユーザのさまざまな情報データとして、目的に対応して例えば、ユーザの心拍、血液情報の生態情報（健康状態の診断管理）、写真画像（写っている人、物のタグ付け）、手書き文字の画像データ（手書き文字認識によるユーザ認証）などがあげられる。

【００６４】

ついで、コンピュータ１７で、作成されたデータベースに基づき、特徴数が定義される

。この特徴数によって、暗号化のための暗号鍵を生成する（ステップＳ１）。

【００６５】

例えば、暗号化として転置式暗号を用い、異なるｎ(2≦n≦N_f!)個の暗号鍵を生成する。ここで、転置式暗号とは、ベクトルの要素の順序をシャッフルすることによる行われる暗号化であり、ベクトル要素の置換である。例えば鍵をK＝(K_1, K_2,…K_m)とするとき、K_i＝jならば、特徴ベクトルのｊの位置をiの位置に置換する。

【００６６】

コンピュータ１７で、作成されたデータベースに基づき、ｎ個のそれぞれの鍵を用いて学習データの暗号化を行う。これによりｎ個の異なる学習データセットが生成される。暗号化された学習データセットをそれぞれ用いて学習を行い、ｎ個の機械学習（ＥＬＭ）モデルを作成する（ステップＳ２）。

【００６７】

この際、中間層における隠れニューロンの重み行列（Ｗ，ｂ）は乱数である、複数のモデルで共有することが出来る。ｎ個のＥＬＭモデルに対して１つの隠れニューロンの重み行列で十分である。一方、出力ニューロンの重みβは、暗号鍵と一対一に対応させる必要があるため、ｎ個生成する必要がある。

【００６８】

このように機械学習のためのニューラルネットワークに使用する中間層の重みＷとバイアスｂ、更に出力層の重みβが生成される。

【００６９】

ついで、求められた重みＷとバイアスｂは、サーバ１０に送られ保存される（ステップＳ３）。

【００７０】

一方、ｎ個の鍵Key_1, Key_2,…Key_nとそれに対応するｎ個の出力層の重みβ$_1$, β$_2$,…β$_n$は、携帯端末１１に送られ保存される（ステップＳ４）。

【００７１】

上記の学習フェーズPh1が終了すると、次に処理は、分類フェーズPh2に移行する。

【００７２】

「分類フェーズPh2」

図４において、携帯端末１１は、ユーザからの情報に基づき、特徴ベクトルｘを作成取得する（ステップＳ５）。

【００７３】

先に学習フェーズPh1でコンピュータ１７から得られたｎ個の暗号鍵から暗号化に用いる一つの鍵をランダムに決定する。ここでは、決定された鍵を鍵ｋとする（ステップＳ６）。

【００７４】

決定された暗号鍵を用いて特徴ベクトルｘを転置式暗号で暗号化する（ステップＳ７）。

【００７５】

【数１０】

$$x'=transposition_k(x) \qquad \cdots (9)$$

【００７６】

次いで、暗号化された特徴ベクトルｘ'を携帯端末１１からサーバ１０に送信する（ステップＳ８）。

【００７７】

サーバ１０では、受信した特徴ベクトルｘ'に対し、演算部１４において、先にステップＳ３において入手している重みＷとバイアスｂを用いて、入力層から隠れ層の計算を先に示した式（２）に従い行う（ステップＳ９）。ついで、演算結果を携帯端末１１に送る（ステップＳ１０）。

【００７８】
携帯端末１１は、演算部１５（図３）において、対応テーブル１３を参照して先に用いた鍵ｋに対応した出力ニューロンの重みβ_kを用いて隠れ層から出力層の演算を、先に示した式（１）に従い実行し結果１６（図３）を出力する。
【００７９】
ここで、上記に説明した例では、隠れ層が一層の例であるが、本発明の適用はこれに限られない。図５は、隠れ層が複数であっても、サーバ１０での演算が可能であることを示す図である。
【００８０】
「実験例」
ここで、上記した本発明に従う察知エージェントの性能評価実験による検証を説明する。
【００８１】
性能評価実験の目的は、第１に本発明の察知エージェントのプロトコルを用いることにより携帯端末１１における計算処理の削減を確認することである。なお、実験では、携帯端末やサーバにおける暗号鍵や重みの読み込み時間は、計測に含まれていない。
【００８２】
１）識別時間の計測：
識別時間とは、携帯端末１１で特徴ベクトルｘを入力してからクラスラベルが出力されるまでの時間である。かかる識別時間を計測することにより携帯端末１１での計算処理を削減できているか否かを確認することが出来る。
【００８３】
携帯端末１１として、実験ではAndroidスマートフォンを用いた。
【００８４】
測定を行ったのは以下に示す３つの手法である。
【００８５】
ａ： All Smartphone:スマートフォン（携帯端末１１）のみでＥＬＭによる識別を行う場合の識別時間の測定である。
【００８６】
ｂ：All Server:ＥＬＭモデルを全てサーバ１０に保存し，スマートフォン（携帯端末１１）からサーバ１０へ送信されたデータを識別，その結果をスマートフォンに返す場合の識別時間の測定である。
【００８７】
ｃ： Cloud System: 本発明で提案したクラウドシステム（プロトコル）を用いて識別を行う場合の識別時間の測定である。
【００８８】
評価実験には，UCI Machine Learning Repository（非特許文献１），および mldata.org（非特許文献２）により公開されているデータベースを用いた。
【００８９】
これらのデータベースの情報を下記の表１に示す。
【００９０】

【表１】

	クラス数 (N_c)	特徴数 (N_f)	データ数 (N_d)
Australian	2	14	690
Satimage	6	36	6,435
USPS	10	256	9,298
Colon cancer	2	2,000	62
Dexter	2	20,000	600
Farm Ads	2	54,877	4,143

【００９１】
データベースAustralian, Satimage, USPS, Colon cancerにおいて平均が０，分散が１になるような正規化を行った．Dexter, Farmadsに関してはスパース性を持つデータベースであるため正規化を行わなかった(正規化は時間測定の結果には影響を与えないため，フェアな比較である)。
【００９２】
ネットワークはスマートフォン（携帯端末１１）１台、サーバ２０を１台，ルータ１台の計３台で構成した．スマートフォン，ルータ間は無線，サーバ，ルータ間はLANケーブルで接続した。
【００９３】
表２に、スマートフォン（携帯端末１１）の仕様情報を示し、表３に、サーバ１０の仕様情報を示す。ルータは NEC PA-WG1800HP2を用いた。
【００９４】
【表２】

Machine	Google Nexus6
OS	Android 5.1 ART VM
CPU	Qualcomm 2.7GHz quad-core krait 450
Memory	3 GB
Wi-Fi	IEEE802.11b/g/n

【００９５】
【表３】

Machine	Dell Precision-WorkStation-T3400
OS	Ubuntu 12.04
CPU	Intel Core2 Duo E8500 (3.16GHz)
Memory	4GB

【００９６】
ＥＬＭに関しては，Python 3.4.3とそのオープンソースライブラリであるNumpy 1.9.2(非特許文献３)，SciPy 0.15.1（非特許文献４)，Scikit-learn 0.16.1(非特許文献５、６）を用いて実装した。
【００９７】
活性関数は標準シグモイド関数，隠れニューロン数N_hは100，500，1,000とした。スマートフォン(携帯端末１１)，サーバ１０間の通信はHTTP通信を行い，データはJSON形式にシリアライズして行った。
【００９８】
サーバ１０はPythonWebフレームワークであるbottle 0.12.8（非特許文献７）を用いた。携帯端末１１であるAndroidスマートフォンでは HTTP 通信ライブラリ OkHttp 2.4.0(

非特許文献８)，JSON ライブラリ Jackson 2.6.0-rc3（非特許文献９）を用いた。

【００９９】

「実験結果と考察」

最初に、十分なＥＬＭの性能を出すためのN_hを検証するため交差検定による実験を行った。データベース，及び隠れニューロン数N_h毎の１０回の５分割交差検定による識別率の平均値(%)を表４に示す。

【０１００】

【表４】

	N_h		
	100	500	1,000
Australian	**85.0**	62.0	54.9
Satimage	81.9	88.3	**91.2**
USPS	75.5	87.2	**89.9**
Colon cancer	74.4	84.8	**86.6**
Dexter	63.0	55.5	**71.2**
Farm-ads	74.8	82.8	**83.2**

【０１０１】

表４において、太字は，各データベースにおける最も高い精度を表している。いくつかのデータベースにおいて、N_h=1,000のとき高い精度を示していた。しかし、Australianにおいては N_h=100のとき，最も高い精度だった。したがって、高い精度を得るためには、各データベースによって最適な隠れニューロン数N_hを選択する必要がある。

【０１０２】

図６－図１１にデータベース毎の識別時間の評価実験による結果を示すグラフである。すなわち、それぞれ、上記のAustralianデータベース、Satimageデータベース、USPSデータベース、Colon cancerデータベース、Dexterデータベース、更にFarm-adsデータベースを使用した場合の識別時間の比較を示している。

【０１０３】

図６－図１１において、共通にグラフの縦軸は識別時間（秒)，横軸は隠れニューロン数N_hであり、３０回の識別時間測定の平均値を表している。それぞれのグラフの縦棒は左から順にAll Smartphone，All Server，Cloud System（プロトコル）における結果である。それぞれのグラフにおいて、エラーバー（ＥＢ）は各測定結果の標準偏差を表している。

【０１０４】

図６－図１１から特徴数N_f，隠れニューロン数N_hが大きい値であるほど、いずれのデータベースを使用する場合でも識別に時間がかかっていることが示される。式（１)の計算量は，特徴数N_f，隠れニューロン数N_hに依存しており、N_f，N_hの値が増加するにつれて、計算量も増加するからである。

【０１０５】

また、特徴数N_fが大きいほど、特徴ベクトルのデータ容量が大きくなり，スマートフォンからサーバへの転送時間が増加する。したがって、いずれの手法においても，特徴数N_fと識別時間，隠れニューロン数N_hと識別時間の間には、正の相関関係があると考えられる。

【０１０６】

また、いずれの隠れニューロン数においても、Cloud SystemよりAll Serverの方が識別時間が速かった。サーバからスマートフォンへの式（２）の計算結果の送信や、スマートフォンにおける式（１）の計算に時間が必要であるためである。しかし、All Serverは全ての識別処理をサーバ１０で行っているため，サーバの情報漏洩の際、第三者に有意な情

報が渡る可能性があり、プライバシーの安全性に欠ける。

【０１０７】

All ServerはCloud Systemよりも高速に識別を行うことができるが、実用性が低い。データベースAustralian, Satimage, USPS, Colon cancerの実験結果に関しては，いずれの隠れニューロン数においても、識別時間は １）All Smartphone, ２）All Server, ３）Cloud Systemの順に速かった。

【０１０８】

一方、図８、図９に示すように、Dexter, Farm-adsの実験結果に関しては １）All Server, ２）Cloud System, and ３）All Smartphoneの順であった。スマートフォンにおける式（１）の計算時間がクラウドサーバとの通信時間を上回るためだと考える。携帯端末のみで，特徴数が比較的多いデータを，高い精度で識別することは困難であることを示している。本実験の結果から、データの特徴数が比較的多い場合、本発明に従うクラウドシステム（プロトコル）を用いることで高速に識別が行えることが確認できた。

【０１０９】

したがって、携帯端末に察知エージェントを搭載する場合、携帯端末における識別コストとプライバシーの保護を考慮すると、本発明の有用性は十分にある。

【０１１０】

[第２の実施例]

ここで、本発明の第２の実施例として、分類処理の安全性を更に高めるため、３つの鍵を用いたプロトコル手続きについて以下に説明する。この実施例では、上記した実施例に対して更に次の特徴利点を有する。

【０１１１】

サーバ１０から携帯端末１へのレスポンスの情報も守られる。これによりさらなる安全性の向上が期待される。サーバ１０の処理に乱数行列を用いる。

【０１１２】

図１２は、第２の実施例に対応するニューラルネットワークであり、携帯端末１１側にサーバ１０における隠れ層計算結果出力に対してドロップ処理機能部３０が付加され、出力層の重みβを変えるだけで隠れ層の重み行列とバイアスが共通に複数のユーザに使用されることを示している。

【０１１３】

すなわち、第２の実施例では、十分大きな乱数行列（ランダムマトリクス）からサブマトリクスを通して部分行列を決定する。これにより一つの乱数行列を複数の察知エージェント、複数のユーザが共有して使うことができる。そのため、サーバ１０は、エージェントやユーザ毎に異なる行列を持つ必要がなく、ただ一つの乱数行列を保持していれば良い。これにより、異なるエージェント、ユーザ毎に乱数行列を読み込み直す必要がないために、サーバ１０の処理の高速化や実装コストの削減が期待できる。

【０１１４】

かかる第２の実施例における３つの鍵を１）置換鍵（trans-key），２）部分行列の位置の鍵(pos-key)，３）隠れ層の出力ベクトルから要素を選択するための鍵（drop-key）と定義する。３つの鍵のそれぞれの役割は、次のようである。

（trans-key）：

ベクトル置換によるユーザからのデータを暗号化するための鍵である。

【０１１５】

これにより暗号化されユーザのデータが保護される。置換鍵は、置換後の要素の添字を並べたベクトルになる。

【０１１６】

暗号化したい特徴ベクトルがv、置換鍵が$K_t=(k_1, k_2, \cdots k_{Nf})$($N_f$はvの次元数)の場合を想定する。暗号化の処理は、$k_c=d$の時($1 \leqq c \leqq N_f$)，vのd番目の要素が暗号化後のc番目のベクトルの要素になる。例として、$v=(v_1, v_2, v_3)$，$K_t=(2, 3, 1)$の時、暗号化後のc番目のベクトルは、$Enc(v, K_t)=(v_2, v_3, v_1)$となる。

【０１１７】
(pos-key)：
サーバ１０に保存された十分大きい乱数行列Woから部分行列W（入力層から隠れ層間の重み）を決定するための鍵である。鍵は、行番号と列番号の２要素のベクトルになる。なお、行列Wのサイズは、

$$N_h' \times N_f$$

である。

$$N_h'$$

は、後に説明する。例えば、pos-keyがKp＝(e,f)のとき、Woのe行f列目の要素が、Wの１行１列目の要素となるようにする。したがって、Woのサイズが、γ×δのとき、Kp=(e,f)の各要素の範囲は、1≦e≦γ－N_h，１≦f≦δ－N_fである。
（drop-key）：
隠れ層の出力ベクトルからその後の処理に使用しない要素を決定する鍵である。(以降この処理を「ドロップする」と表記する）この処理の意図は、全ての隠れ層の出力を、その後の計算で使わせなくさせることである。サーバ１０の計算の出力結果のうち、実際に計算に使われる要素を隠すことができるので、安全性の向上が果たされる。
【０１１８】
drop-key K_dは隠れ層の出力のうち、ドロップする要素の添字を並べたベクトルになる。
【０１１９】
【数１１】

$$N_h' = r \cdot N_h$$

【０１２０】
が隠れ層の出力ベクトルの次元数のとき、K_dの各要素の範囲は、
【０１２１】
【数１２】

$$1 \leq k_d \leq N_h'$$

【０１２２】
であり、大きさは(r－1)・N_hになる。rは、隠れニューロン数の冗長率である。ここでの冗長とは、隠れニューロン数N_hを必要より多く設定し、隠れ層の計算を余分に行うことである。
【０１２３】
rの範囲は、
【０１２４】
【数１３】

$$[1, t],\ t \in R$$

【０１２５】
である。真の隠れニューロン数がN_hのとき、プロトコルでは
【０１２６】
【数１４】

$$N_h' = r \cdot N_h$$

【０１２７】
で隠れ層までの処理を行う。例えば、真の隠れニューロン数N_h＝２、冗長数r＝２．０のとき、

【０１２８】
【数１５】

$$N_h' = r \cdot N_h = 2.0 \cdot 2 = 4$$

【０１２９】
になる。したがって、プロトコルは隠れニューロン数は、
【０１３０】
【数１６】

$$N_h' = 4$$

【０１３１】
として隠れ層の計算を行う。
【０１３２】
【数１７】

$$N_h' = 4$$

【０１３３】
なので、隠れ層の出力は、$h(v)=(h_1,h_2,h_3,h_4)$, $K_d=(1, 3)$のとき、処理後の出力は、$drop(h(v),K_d)=(h_2,h_4)$になる。
【０１３４】
このように、ドロップ処理により、隠れ層の計算を必要以上に行った分を真に必要な分に調整することになる。
【０１３５】
つぎに、上記３つの鍵の処理を加えた第２の実施例のプロトコル処理について処理フローに従い説明する。
【０１３６】
図１３は、第２の実施例に対応する学習(トレーニング)の処理フローである。
【０１３７】
ここで、ローカルサーバ（ＰＣ）１７は、各ユーザが持つコンピュータ（パソコン）であって、本発明のプロトコルを用いて察知エージェントを実運用する際、第１の実施例と同様に学習処理はユーザの所有するコンピュータ１７で行うことを想定する。
【０１３８】
まず、コンピュータ１７により、十分大きい乱数行列Woを生成定義する(ステップＳ２０)。なお、前記Woのサイズは任意であり、任意の分類タスク、及び複数のユーザで共有して使うことが出来る。そのため、Woは一度だけ生成すればよい。
【０１３９】
携帯端末１１によりトレーニングデータを収集し、コンピュータ１７に送る（ステップＳ２１）。
【０１４０】
コンピュータ１７では、冗長度r、真の隠れニューロン数N_hを決定し、
【０１４１】
【数１８】

$$N_h' = r \cdot N_h$$

【０１４２】
を求める(ステップＳ２２)。
【０１４３】
次いで、コンピュータ１７は、
【０１４４】

【数１９】

n個のtrans-keys $K_t^1, K_t^2, \ldots K_t^n$ ， m個のpos-keys $K_p^1, K_p^2, \ldots K_p^m$ ，l個の $K_d^1, K_d^2, \ldots K_d^l$

【０１４５】

を順次生成する(ステップＳ２３－１，２３－２，２３－３)。

【０１４６】

ついで、暗号化データと隠れ層から出力層の重みβを、以下のように算出する(ステップＳ２４)。

【０１４７】

ｎ個のtrans-keyそれぞれを使って学習データセットを暗号化する。すなわち、ｎ個の異なる学習データセットが生成される。

【０１４８】

例えば、学習データ数が、N_d，N_f＝４の時、学習データセットは、式（１０）である。特徴ベクトルを行に並べたもの、あるいは、trans-keyが式（１１）であるとき、暗号化されたデータセットは、式（１２）になる。

【０１４９】

【数２０】

$$X=\begin{pmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N_d1} & x_{N_d2} & x_{N_{d3}} & x_{N_d4} \end{pmatrix}^t \quad (10)$$

【０１５０】

【数２１】

$$K_t=(4\ 3\ 1\ 2) \quad (11)$$

【０１５１】

【数２２】

$$X' = \text{transposition}(X)$$

$$=\begin{pmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N_d1} & x_{N_d2} & x_{N_{d3}} & x_{N_d4} \end{pmatrix}^t \quad (12)$$

【０１５２】

Woと複数のpos-keyと

【０１５３】

【数２３】

N_h', N_f

【０１５４】

を使ってｍ個の異なる部分行列（隠れ層の重み）を生成する。

【０１５５】

例えば、Woが式（１３）、pos-keyが式（１４）で、

【０１５６】

【数２４】

$$r = 1.6,\ N_h = 5,\ N_h' = r \cdot N_h = 1.6 \cdot 5 = 8,\ N_f = 4$$

【０１５７】
であるとき、部分行列は式(１５)になる。Ｗの大きさは、
【０１５８】
【数２５】

$$N_f \times N_h' = 4 \times 8$$

【０１５９】
である。
【０１６０】
【数２６】

$$\mathrm{Wo}=\begin{pmatrix} w_{1,1} & \cdots & w_{1,N} \\ \vdots & \ddots & \vdots \\ w_{N,1} & \cdots & w_{N,N} \end{pmatrix} \qquad (13)$$

【０１６１】
【数２７】

$$\mathrm{K_p}=(2 \quad 9) \qquad (14)$$

【０１６２】
【数２８】

$$\mathrm{W}=\mathrm{position}(W_0, Kp, N_h') = \begin{pmatrix} w_{2,9} & \cdots & w_{2,17} \\ \vdots & \ddots & \vdots \\ w_{6,9} & \cdots & w_{6,17} \end{pmatrix} \qquad (15)$$

【０１６３】
ついで、それぞれのデータセット、それぞれの重みを用いて、隠れ層の出力を求める。その出力に対してそれぞれのdrop-keyを用いて複数のドロップされた出力を求める。これにより、ｎ，ｍ，ｌ個の異なる隠れ層の出力が決定する。
【０１６４】
例えば、
【０１６５】
【数２９】

$$\mathrm{r}=1.6,\ N_h=5,\ N_h'=\mathrm{r}\cdot N_h=1.6\cdot 5=8,\ N_f=4$$

【０１６６】
で、隠れ層の出力を式（１６）とする。
【０１６７】
drop-keyが式（１７)のとき、ベクトルの大きさは、(ｒ－１)・N_h＝(１．６－１)・５＝３であって、式（１６）の５，４，７行目をドロップさせるので、結果は式（１８）になる。
【０１６８】

【数３０】

$$H = \begin{pmatrix} h_{1,1} & h_{1,2} & h_{1,3} & h_{1,4} \\ h_{2,1} & h_{2,2} & h_{2,3} & h_{2,4} \\ h_{3,1} & h_{3,2} & h_{3,3} & h_{3,4} \\ h_{4,1} & h_{4,2} & h_{4,3} & h_{4,4} \\ h_{5,1} & h_{5,2} & h_{5,3} & h_{5,4} \\ h_{6,1} & h_{6,2} & h_{6,3} & h_{6,4} \\ h_{7,1} & h_{7,2} & h_{7,3} & h_{7,4} \\ h_{8,1} & h_{8,2} & h_{8,3} & h_{8,4} \end{pmatrix} \qquad (16)$$

【０１６９】
【数３１】

$$K_d = (5\ \ 4\ \ 7) \qquad (17)$$

【０１７０】
【数３２】

$$H' = \mathrm{drop}(H,\ K_d)$$

$$= \begin{pmatrix} h_{1,1} & h_{1,2} & h_{1,3} & h_{1,4} \\ h_{2,1} & h_{2,2} & h_{2,3} & h_{2,4} \\ h_{3,1} & h_{3,2} & h_{3,3} & h_{3,4} \\ h_{6,1} & h_{6,2} & h_{6,3} & h_{6,4} \\ h_{8,1} & h_{8,2} & h_{8,3} & h_{8,4} \end{pmatrix} \qquad (18)$$

【０１７１】
次に、それぞれの隠れ層の出力を使って重みβを求める。これによりｎ，ｍ，ｌ個の異なるβが生成される。
【０１７２】
したがって、K_t，K_p，K_d，β_s更に
【０１７３】
【数３３】

$$N_h'$$

【０１７４】
の全ての携帯端末１１に保存する（ステップＳ２５）。さらに、Woをサーバ１０に保存する（ステップＳ２６）。
【０１７５】
次に本発明の第２の実施例に従う分類処理について説明する。図１４は、かかる分類処理の処理フローである。
【０１７６】
携帯端末１１は、ユーザからデータを取得する（ステップＳ３０）。trans-key K_tを選択し（ステップＳ３１）、選択した
【０１７７】
【数３４】

$$\text{trans-key}\ k_t^i$$

【０１７８】

を用いて取得したユーザデータの特徴ベクトルを暗号化する（式１９）(ステップＳ３２)。

【０１７９】

このとき、iは，1≦i≦nからランダムに定義される。

【０１８０】

【数３５】

$$x'=\text{transposition}\ (x, K_t^i) \qquad (19)$$

【０１８１】

例えば、特徴ベクトルが式（２０）、K_tが式（１１）のとき、暗号化後のベクトルは、式（２１）のようになる。

【０１８２】

【数３６】

$$x = (x_1, x_2, x_3, x_4)^t \qquad (20)$$

【０１８３】

【数３７】

$$x' = \text{transposition}\ (x, Kt) = (x_4, x_3\ x_1, x_2)^t \qquad (21)$$

【０１８４】

携帯端末１１は、

【０１８５】

【数３８】

暗号化された特徴ベクトルx'、pos-key k_p^j、冗長な隠れニューロン数N_h'

【０１８６】

をサーバ１０に送信する(ステップＳ３４)。このとき、ｊは1≦j≦mにランダムに定義される。

【０１８７】

サーバ１０では、

【０１８８】

【数３９】

Wo、pos-key k_p^j、N_h'

【０１８９】

を用いて式（２２）により隠れ層の重みＷを求め(ステップＳ３５)、隠れ層の計算(式（２３))を行う（ステップＳ３６）。

【０１９０】

【数４０】

$$W=\text{position}\ (Wo,\ k_p^j,\ N_h') \qquad (22)$$

【０１９１】

【数４１】

$$h(x') = G(W \cdot x') \qquad (23)$$

【０１９２】

ついで、サーバ１０は、計算出力である隠れ層h(x')を携帯端末に送信する（ステップＳ３７）。

【０１９３】

携帯端末１１において、
【０１９４】
【数４２】

$$\text{drop-key } K_d^k$$

【０１９５】
をもとに、後の計算に使う要素を選択する(ステップＳ３８：式（２４））。このとき、ｋは、1≦k≦1からランダムに定義される。
【０１９６】
【数４３】

$$h_{drop} = drop(h(x'),\ K_d^k) \qquad (24)$$

【０１９７】
例えば、
【０１９８】
【数４４】

$$r = 1.6,\ N_h = 5,\ N_h' = r \cdot N_h = 1.6 \cdot 5 = 8,\ N_f = 4$$

【０１９９】
で、隠れ層の出力を式（２５）とする。K_dが式（１７）のとき、結果は式（２６）となる。
【０２００】
【数４５】

$$y = (h_1\ h_2\ h_3\ h_4\ h_5\ h_6\ h_7\ h_8)^t \qquad (25)$$

【０２０１】
【数４６】

$$y' = drop(y, K_d) = (h_1\ h_2\ h_3\ h_6\ h_8)^t \qquad (26)$$

【０２０２】
携帯端末１１は、β_{ijk}を用いて出力層の計算をし、結果を得る。
【０２０３】
【数４７】

$$label = \text{sign}\,(h_{drop} \cdot \beta_{ijk})$$

【０２０４】
上記第２の実施例でのプロトコルでは、暗号化した状態で最終結果を得るため、復号化の処理を行う必要はない。
【０２０５】
以下に、上記第２の実施例における効果を確認するために行った実験例について説明する。
【０２０６】
[実験例１]
この実験での目的は、各データセットに対し十分な性能を出すための最適なN_hを探すことにある。
【０２０７】
評価は、５分割交差検定による識別正答率で行う。なお、５分割交差検定とは、「データセットを特徴ベクトルの数で５分割し、１／５をテスト用データ、４／５を学習用データとして使う。そして、テストデータ、学習データを異なる５パターンの組み合わせのそ

れぞれで性能評価を行い、その５回分の平均値を最終結果とする。」機械学習アルゴリズムの分類性能を評価するための一手法であり、一般にｋ分割交差検定と呼ばれる。

【０２０８】

かかる５分割交差検定において、最適なN_hを探すため、N_h＝1000，1500，2000,…5000とN_hを変え、それぞれの値で５分割交差検定を行った。活性化関数には、標準シグモイド関数を用いた。なお、実験は、Python3，Numpy 1.10.4（非特許文献３），Scipy 0.17.0（非特許文献４），及びScikit-learn 0.18.1（非特許文献５）を用いて本発明者が作成したプログラムによって実行した。

【０２０９】

[実験例２]

この実験は、本発明のプロトコルのシミュレーション実験である。以下の手法それぞれで分類時間の測定を行い、比較を行った。

【０２１０】

１）Local:ＥＬＭによる全ての分類処理を携帯端末１１のみで行う。

【０２１１】

２）Protocol：本発明によるプロトコルにより行う。

第３の発明のプロトコルが有効であるかを確認することが出来る。

【０２１２】

真の隠れニューロン数には実験例１で得られた最適な(高い性能が出る)値を設定した。trans-keyの数ｎ，pos-keyの数ｍ，drop-keyの数ｌは、それぞれ５に設定した。ｒ＝１．４に設定した。Woのサイズは、10,000×10,000で行った。

【０２１３】

携帯端末１１として２つのAndroidスマートフォンを用いてそれぞれで実験を行った。

【０２１４】

各実験でのネットワークは、スマートフォン１つ、ルーター１つ、サーバ１つで構成した。スマートフォンとルータは無線（ＷｉＦｉ）で接続され、サーバとルータはＬＡＮケーブルで接続された。

【０２１５】

実験に用いた携帯端末１１としてのスマートフォンとサーバ１０の情報は、それぞれ以下の表５，表６に示すとおりである。

【０２１６】

【表５】

スマートフォンの性能

Machine	Google Nexus6	Motorola Moto Z play
OS	Android 7.0	Android 7.0
Chipset	Snapdragon 805	Snapdragon 625
CPU	Quad-core 2.7 GHz	Octa-core 2.0 GHz
	Krait 450	Cortex-A53
Memory	3 GB	3 GB
Wi-Fi	IEEE 802.11 b/g/n	IEEE 802.11 ac/a/b/g/n

【０２１７】

【表６】

サーバの性能と環境

Machine	Dell Precision-WorkStation-T3400
OS	Ubuntu 12.04
CPU	Intel Core2 Duo E8500 (3.16GHz)
Memory	4GB

【０２１８】
プロトコルのためのシステムはWeb ＡＰＩとして実装され、通信には、ＨＴＴＰを用いた。

【０２１９】
ＥＬＭのモデル（Wといくつかのβ）と３つの各鍵（K_t, K_p, K_d）は、Python3で生成し、npyファイルフォーマットでダンプした。なお、npyファイルフォーマットとは、Numpy arrayの標準的なバイナリーフォーマットである。かかるＥＬＭも本発明者により著作した。

【０２２０】
サーバ１０は、Python3 bottle(非特許文献７),Nginx(非特許文献１４),及びuWSGI（非特許文献１５）で実装された。

【０２２１】
Androidスマートフォンでのプログラムは、Java（登録商標）とC++で書かれた。Javaは、Android開発のための標準的な開発言語である。Android端末でのサーバ１０と通信を行う部分は、Rxjava(ReactiveX)(非特許文献１６)、Retrfit(非特許文献１７)、Gson(非特許文献１８)をライブラリとして使用した。

【０２２２】
サーバ１０へは、JSONフォーマットでデータを送信するようにした。また、Android端末上での重み（W,β）や鍵（K_t, K_p, K_d）の読み込み、行列演算の処理にはAndroid NDKを用いた。

【０２２３】
Android NDK(C++)で実装することにより、Javaでの実装より高速に実行することが出来る。npyファイルフォーマットの読み込み、及び行列演算の処理をC++で記述した。C++プログラムとJavaプログラムの接続は、Java Native Interface(JNI)を用いた。行列積には基本的な3-loopの方法を用いた。

【０２２４】
実験において、モデルの読み込み、プロトコルの処理、モデルのメモリからの削除の３つの測定範囲を含め、１つの分類処理を３０回行い、その平均時間を求めた。

【０２２５】
ここで、実験に用いたデータセットについて説明する。実験に用いたデータセットは表７に示すとおりである。

【０２２６】

【表７】

データセット

	class (N_c)	Features (N_f)	Data Count(N_d)
Australian	2	14	690
QSAR	2	41	1,055
Gisette	2	5,000	7,000
Satimage	6	36	6,435

【０２２７】

上記データセットのそれぞれについて説明する。

１）Australian:

Australian(非特許文献２０)は、クレジットカードを認可するか否かを判断するためのデータである。それぞれの特徴ベクトル（N_f）は、一人分のデータを表す。クラスラベル（N_c）は高信用か低信用かの二値である。

２）QSAR:

QSAR(非特許文献２１)は、定量的構造活性相間に関するデータである。

３）Gisette:

Gisette(非特許文献２２)は、手書き数字画像分類問題であり、“４”と“９”のどちらかを分類する。

４）Satimage:

Satimageは、地球観測衛星（LANDSAT)による写真に基づいたデータである。クラスラベルは以下の６つの風景である。１．赤い土壌（red soil）２.綿花（cotton crop） ３．灰色の土壌（grey soil）４．湿った灰色の土壌(damp grey soil) ５．植物の切り株がある土壌（soil with vegetation stubble） ６．より湿った灰色の土壌（very damp grey soil）

【０２２８】

[実験結果]

上記第２の実施例に対する実験１、実験２の結果をまとめると次のようである。

【０２２９】

実験１の結果：

図１５Ａ－１５Ｄは、上記４つのデータセットのそれぞれにおける１０回の５分割交差検定による正答率を表している。縦軸は正答率、横軸は隠れニューロン数を示している。“Train”ラベルは学習データでのスコアを示す、“Test”ラベルはテストデータでのスコアを示している。

【０２３０】

図１５Ａ，図１５ＢにそれぞれAustralianとQSARの結果において過学習が起きている。すなわち、学習モデルが、学習データに依存しすぎ、学習データは高性能に分類できるがテストデータに関しての分類において精度がなくなる状態である。

【０２３１】

これらの結果による各データセットの最適なN_hを示すと表８に示すようである。

【０２３２】

【表８】

最も良い性能が出るパラメータ N_h

Dataset	N_h	Acc.
Australian	100	79.8
QSAR	60	85.5
Gisette	2,000	97.0
Satimage	2,500	94.6

【０２３３】

実験２の結果：

図１６Ａ－図１６Ｄ、及び図１７Ａ－図１７Ｄは、上記４つのデータセットのそれぞれでの分類処理時間を示している。処理時間が小さい値ほど高速の処理が行われることを示している。

【０２３４】

Nexus6での結果は図１６Ａ－図１６Ｄ、Moto Z playの結果は図１７Ａ－図１７Ｄである。

【０２３５】

各図において、左側の棒は、携帯端末１１のみを使用する手法での結果、右側の棒は、本発明のプロトコルによる結果を示している。AustralianとQSARは、先に説明のとおり過学習が起こり精度がでないため、N_h=1000までにとどめている。縦軸は分類時間で、単位は、秒である。斜線部分４０は、携帯端末サーバ１１がサーバ１０へリクエストを送り、レスポンスが還ってくるまでの時間である。

【０２３６】

表９、表１０に１回の分類での各処理に要した時間を示す。

【０２３７】

Preとは、trans-keyの読み込みや暗号化、drop-keyの読み込み処理であり、サーバ１０とネットワークの処理時間である。Convertは、Javaオブジェクトをサーバ１０へ送信するためJSONフォーマットに変換する処理である。逆にサーバ１０からのレスポンスをJSONフォーマットからJavaオブジェクトに変換する処理も含む。

【０２３８】

Hiddenは、サーバにおける処理に要した時間である。すなわち、携帯端末でリクエストをしてレスポンスの返ってくる時間である。

【０２３９】

さらに、Outputは、携帯端末１０でのドロップ処理と出力層の計算を行う処理である。また、TotalはProtocol，Localそれぞれの分類処理の合計時間である。

【０２４０】

【表９】

Nexus6 での各処理にかかった時間 (s)

	Protocol					Local
	Pre	Transmission		Output	Total	Total
		Convert	Hidden			
Australian	0.0023	0.0238	0.0101	0.0060	0.0422	0.0026
QSAR	0.0023	0.0248	0.0120	0.0055	0.0446	0.0026
Gisette	0.0151	0.4704	0.1068	0.0145	0.6068	0.9096
Satimage	0.0026	0.1252	0.0174	0.0040	0.1493	0.0124

【０２４１】

【表１０】

Moto Z play の各処理にかかった時間 (s)

Dataset	Protocol					Local
	Pre	Transmission		Output	Total	Total
		Convert	Hidden			
Australian	0.0004	0.0152	0.0056	0.0004	0.0216	0.0005
QSAR	0.0003	0.0134	0.0073	0.0004	0.0214	0.0005
Gisette	0.0025	0.2884	0.0763	0.0013	0.3685	0.7164
Satimage	0.0004	0.0626	0.0121	0.0019	0.0770	0.0089

【０２４２】

[考察]

図１６Ａ－１６Ｄ，及び図１７Ａ－１７Ｄにより全ての結果において、処理時間が１秒以下であった。これはユーザにとっては許容範囲の処理時間である。Nexus6及びMOTO Z playのいずれの携帯端末においてもAustralian, QSAR, Satimage データセットはLocalの方が高速であった。上記表９、表１０によれば、プロトコルはJSONとJavaオブジェクトの返還に時間が掛かっている。よって、変換処理を改善することにより本発明のプロトコル全体の改善が期待できる。

【０２４３】

Gisetteデータセットは両方の端末において、本発明プロトコルの方が高速に動作している。

【０２４４】

プロトコルがサーバ１０で行う隠れ層の計算は、式（２）よりN_hとN_fに依存する。GisetteのようなN_h, N_fともに大きい値の分類処理において有効である。これらの点から本発明に従うプロトコルは携帯端末１１での分類処理を削減する方法として有効である。

【符号の説明】

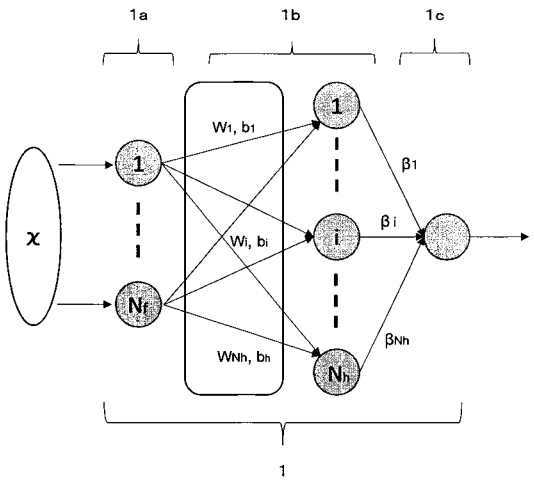
【０２４５】

１　ニューラルネットワーク

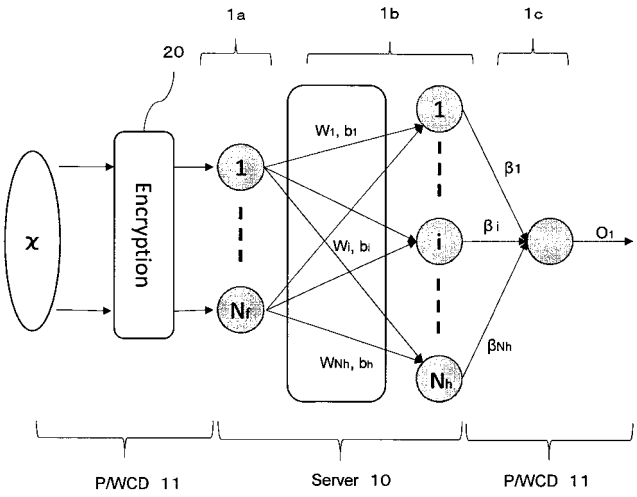
１ａ　入力層

１ｂ　隠れ層（中間層）
１ｃ　出力層
１０　サーバ
１１　携帯端末
１２　無線通信機能
１３　暗号鍵と重みβの対応テーブル
１４、１５　演算部
１６　クラスラベル
１７　コンピュータ
２０　暗号化部
３０　ドロップ処理機能部

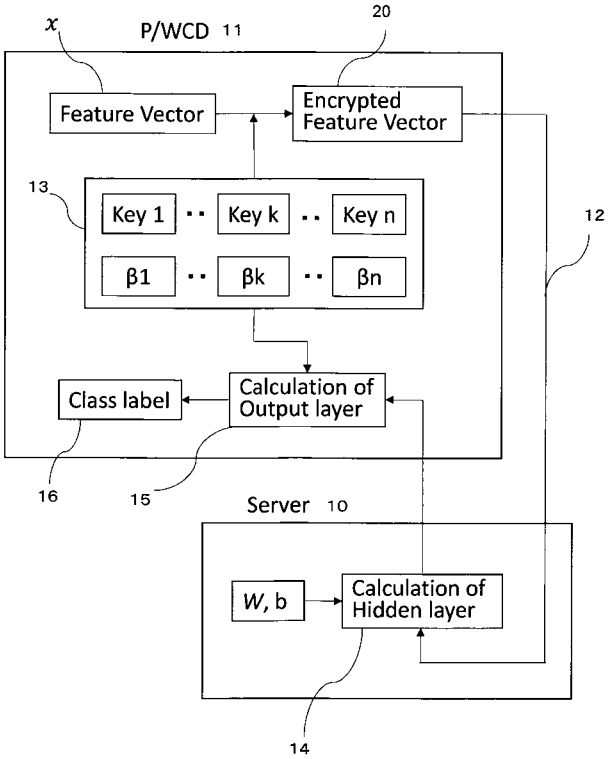
【図１】



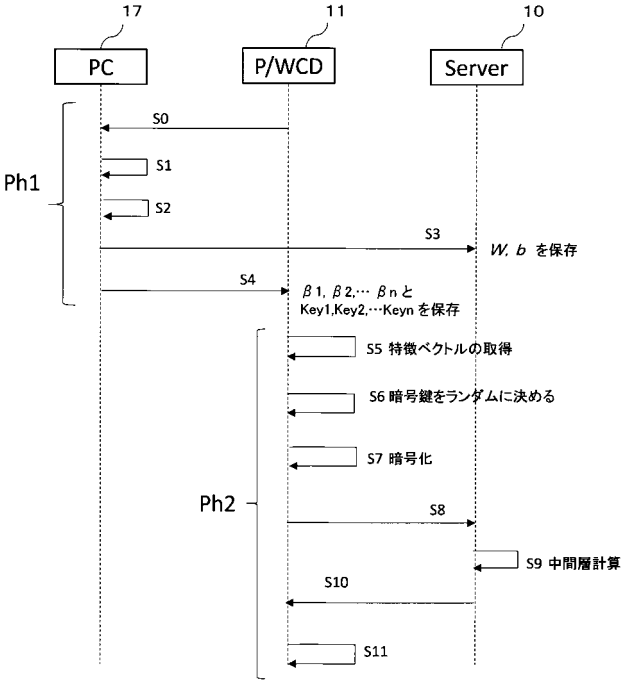
【図２】



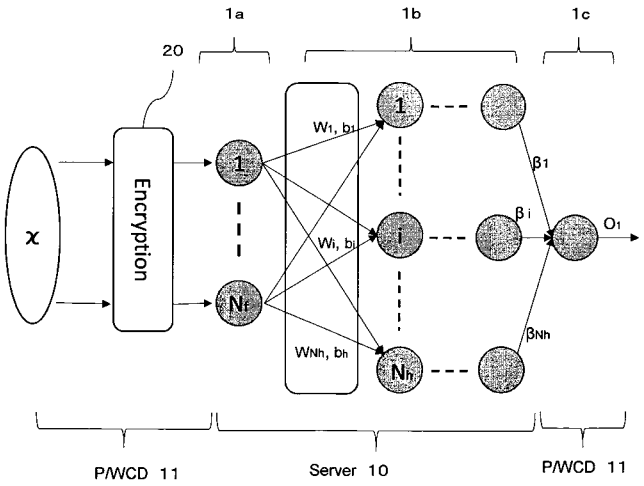
【図３】


x
P/WCD 11
20
Feature Vector
Encrypted Feature Vector
13
Key 1
Key k
Key n
β1
βk
βn
12
Class label
Calculation of Output layer
16
15
Server 10
W, b
Calculation of Hidden layer
14

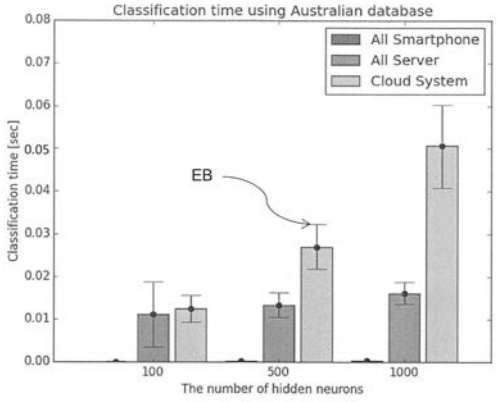
【図４】


17
11
10
PC
P/WCD
Server
S0
S1
S2
Ph1
S3
W, b を保存
S4
β1, β2,… βn と
Key1,Key2,…Keyn を保存
S5 特徴ベクトルの取得
S6 暗号鍵をランダムに決める
S7 暗号化
Ph2
S8
S9 中間層計算
S10
S11

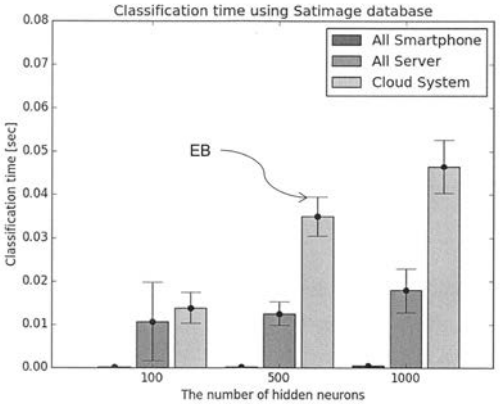
【図５】


1a
1b
1c
20
x
Encryption
1
N
W1, b1
Wi, bi
WNh, bh
1
i
Nh
β1
βi
βNh
O1
P/WCD 11
Server 10
P/WCD 11

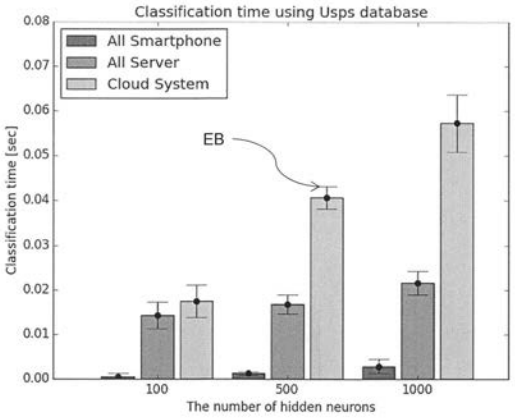
【図６】


Classification time using Australian database
All Smartphone
All Server
Cloud System
EB
Classification time [sec]
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
100
500
1000
The number of hidden neurons

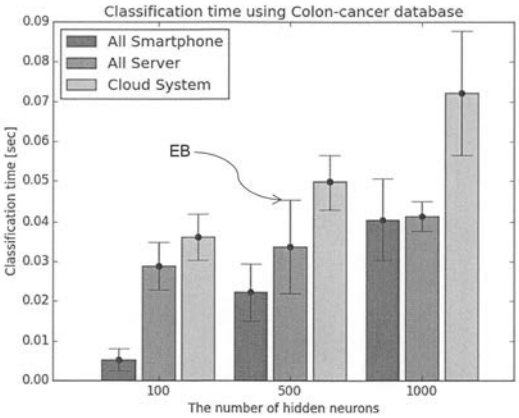
【図７】


Classification time using Satimage database
All Smartphone
All Server
Cloud System
EB
Classification time [sec]
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
100
500
1000
The number of hidden neurons

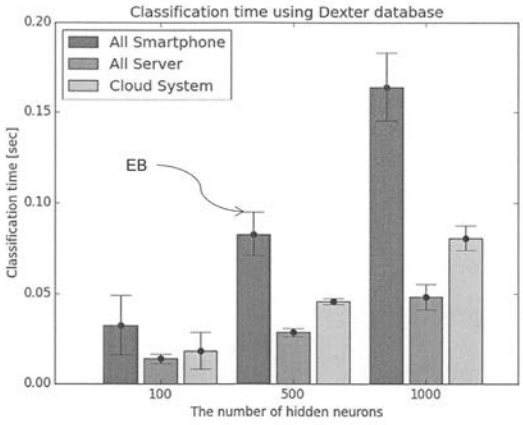
【図８】



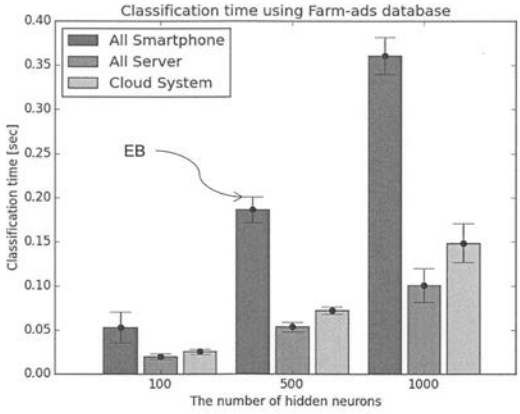
【図９】



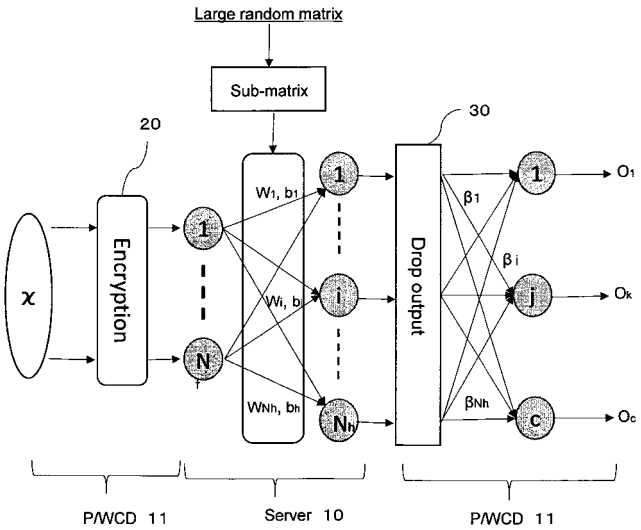
【図１０】



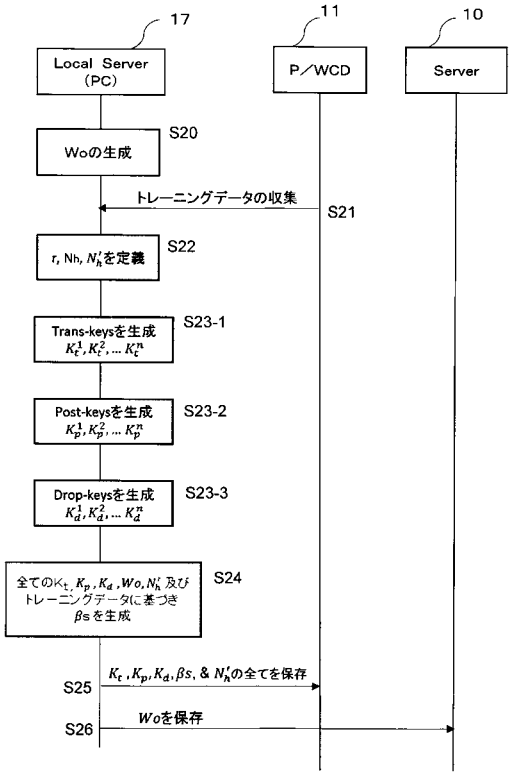
【図１１】



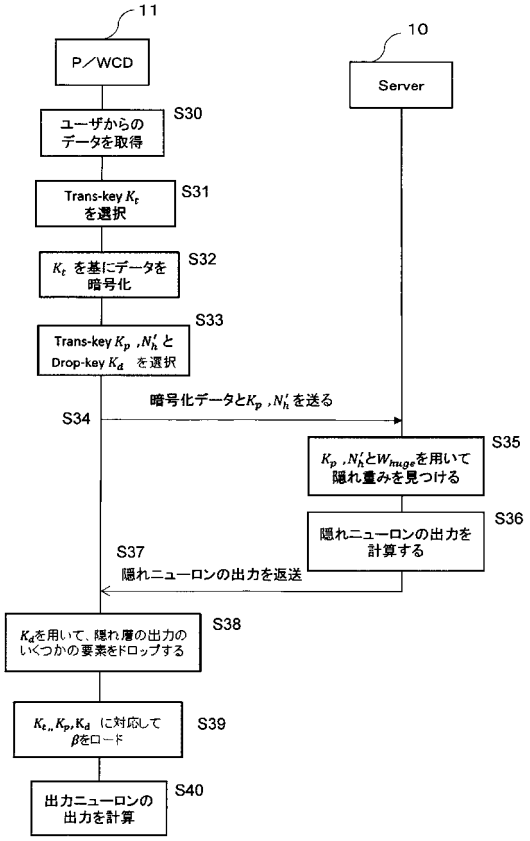
【図１２】



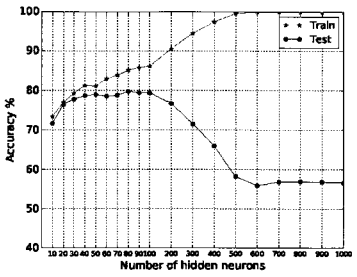
【図１３】



【図１４】

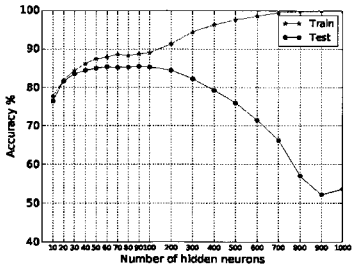


【図１５Ａ】



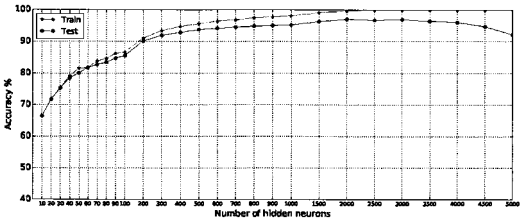
Australian データセットでの正当率

【図１５Ｂ】



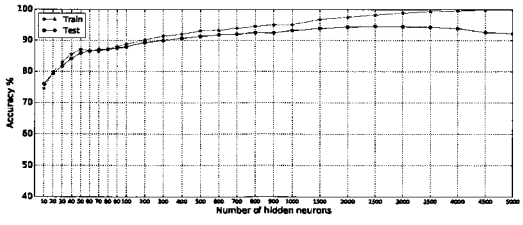
QSAR データセットでの正当率

【図１５Ｃ】



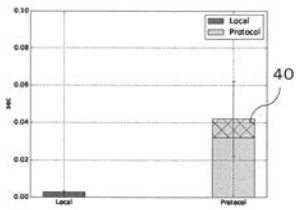
Gisette データセットでの正当率

【図１５Ｄ】



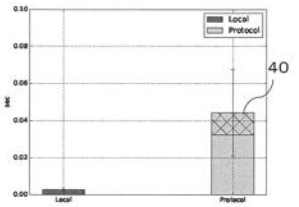
Satimage データセットでの正当率

【図１６Ａ】



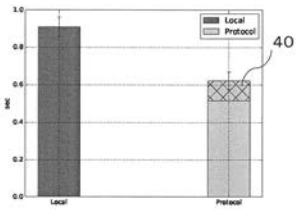
Australian classification time on Nexus6

【図１６Ｂ】



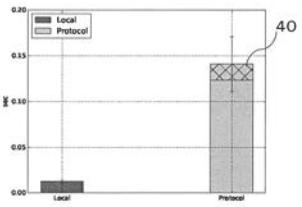
QSAR classification time on Nexus6

【図１６Ｃ】



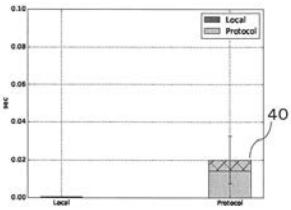
Gisette classification time on Nexus6

【図１６Ｄ】



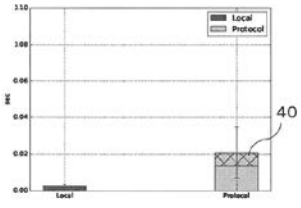
Satimage classification time on Nexus6

【図１７Ａ】



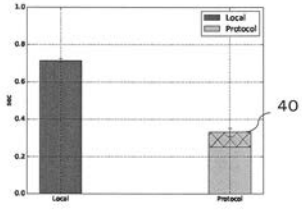
Australian classification time on Moto Z play

【図１７Ｂ】



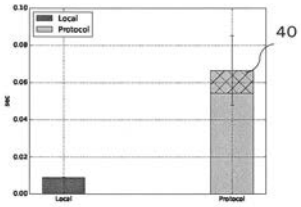
QSAR classification time on Moto Z play

【図１７Ｃ】



Gisette classification time on Moto Z play

【図１７Ｄ】



Satimage classification time on Moto Z play