

(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2024年1月4日(04.01.2024)



(10) 国際公開番号
WO 2024/004212 A1

(51) 国際特許分類:
G06N 3/02 (2006.01)

(21) 国際出願番号: PCT/JP2022/026498

(22) 国際出願日: 2022年7月1日(01.07.2022)

(25) 国際出願の言語: 日本語

(26) 国際公開の言語: 日本語

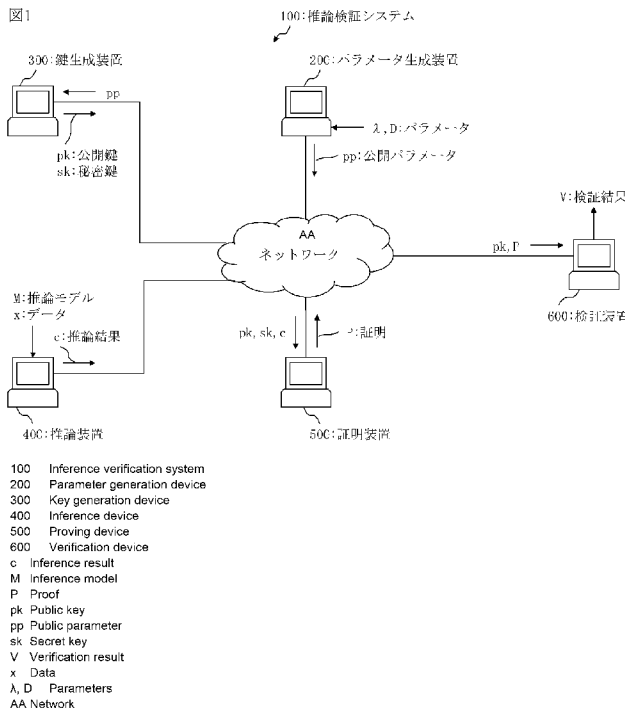
(71) 出願人:三菱電機株式会社(MITSUBISHI ELECTRIC CORPORATION) [JP/JP]; 〒1008310 東京都千代田区丸の内二丁目7番3号 Tokyo (JP).

(72) 発明者: 西田 悠太郎 (NISHIDA, Yutaro); 〒1008310 東京都千代田区丸の内二丁目7番3号 三菱電機株式会社内 Tokyo (JP). 安田 聖(YASUDA, Satoshi); 〒1008310 東京都千代田

区丸の内二丁目7番3号 三菱電機株式会社内 Tokyo (JP). 小関 義博(KOSEKI, Yoshihiro); 〒1008310 東京都千代田区丸の内二丁目7番3号 三菱電機株式会社内 Tokyo (JP). 花岡 悟一郎(HANAOKA, Goichiro); 〒1350064 東京都江東区青海2-3-26 国立研究開発法人産業技術総合研究所内 Tokyo (JP). アッタラパドゥン ナッタポン(ATTRAPADUNG, Nuttapong); 〒1350064 東京都江東区青海2-3-26 国立研究開発法人産業技術総合研究所内 Tokyo (JP). 坂井 祐介(SAKAI, Yusuke); 〒1350064 東京都江東区青海2-3-26 国立研究開発法人産業技術総合研究所内 Tokyo (JP). シュールツ ヤコブ クロイス ナカムラ(SCHULDT, Jacob Chroeis Nakamura); 〒1350064 東京都江

(54) Title: INFERENCE VERIFICATION SYSTEM AND INFERENCE VERIFICATION METHOD

(54) 発明の名称: 推論検証システムおよび推論検証方法



(57) Abstract: An inference device (400) represents a fractional value, which is data subject to an inference process, as an integer value and treats the integer value as a parameter of a convolutional neural network to execute an inference model, to thereby obtain an inference result. A proving device (500) inputs the inference result to execute a proof generation algorithm, to thereby obtain a proof. A verification device (600) inputs the proof to execute a verification algorithm, to thereby obtain a verification result.

東区 青海 2 - 3 - 2 6 国立研究開発法人
産業技術総合研究所内 Tokyo (JP).

(74) 代理人: 弁理士法人クロスボーダー特
許事務所 (CROSS-BORDER PATENT FIRM);
〒2470056 神奈川県鎌倉市大船二丁目 1 7
番 1 0 号 3 階 Kanagawa (JP).

(81) 指定国(表示のない限り、全ての種類の国内保
護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA,
BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN,
CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE,
EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU,
ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH,
KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY,
MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ,
NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT,
QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL,
ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG,
US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) 指定国(表示のない限り、全ての種類の広域保
護が可能): ARIPO (BW, GH, GM, KE, LR, LS,
MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG,
ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, RU,
TJ, TM), ヨーロッパ (AL, AT, BE, BG, CH, CY, CZ,
DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT,
LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS,
SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM,
GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

添付公開書類:

一 国際調査報告 (条約第21条(3))

(57) 要約: 推論装置 (4 0 0) は、推論処理の対象となるデータである小数値を整数値で表現し、前記整
数値を畳み込みニューラルネットワークのパラメータとして扱って推論モデルを実行して推論結果を得
る。証明装置 (5 0 0) は、前記推論結果を入力にして証明生成アルゴリズムを実行して証明を得る。
検証装置 (6 0 0) は、前記証明を入力にして検証アルゴリズムを実行して検証結果を得る。

明 細 書

発明の名称：推論検証システムおよび推論検証方法

技術分野

[0001] 本開示は、ゼロ知識証明による推論モデルの検証に関するものである。

背景技術

[0002] ニューラルネットワークによるAI推論技術は、データ分類などの機械学習タスクで大きな成功を収めている。AIは人工知能を略称である。

ニューラルネットワークによるデータ分析を行うためには、事前に大量の学習データを用いて、推論モデルの学習を行う必要がある。この時、学習データを用意することの困難さ及び計算資源の制約などの理由から、ユーザが自身の環境で推論モデルを構築することが困難である場合がある。

こういった観点から、近年ではクラウド上でニューラルネットワークによるデータ分析を提供するサービス（MLaaS）がある。このサービスにより、クライアントは、分析したいデータをクラウド上にアップロードすることで、提供されている推論モデルによる推論を実行できる。そのため、クライアントは、推論モデルの構築にコストを費やす必要がない。

MLaaSは、Machine Learning as a Serviceの略称である。

[0003] MLaaSにおいて、クライアントが分析したいデータを推論モデルの提供者に送信して推論処理を委託する場合、サービス提供者は、クライアントに対して、推論結果が実際に推論モデルによって分析が行われた結果であること、を証明する必要がある。

最も簡単な解決策は、サービス提供者が推論モデル自体を公開することである。しかし、推論モデルはサービス提供者の知的財産であるため、推論モデルをクライアントに開示することは難しい。

[0004] 非特許文献1は、推論モデルによる推論処理が実際に実行されたことを証明することをゼロ知識証明を用いて可能にする方法を提案している。

この方法により、サービス提供者は、クライアントに対して、推論結果が推論モデルによる分析処理によって得られたものであることを証明することが可能になる。

先行技術文献

非特許文献

- [0005] 非特許文献1: z k C N N : Z e r o K n o w l e d g e P r o o f s
f o r C o n v o l u t i o n a l N e u r a l N e t w o r k
P r e d i c t i o n s a n d A c c u r a c y . T i a n y i L
i u , X i a n g X i e , a n d Y u p e n g Z h a n g . 2
0 2 1 .

発明の概要

発明が解決しようとする課題

- [0006] 非特許文献1の方法は、使用されるゼロ知識証明のプロトコルの制約により、推論モデルのパラメータとして整数値しか扱うことができない。
- [0007] 本開示は、パラメータが小数である推論モデルの検証を可能にすることを目的とする。

課題を解決するための手段

- [0008] 本開示の推論検証システムは、
- 推論処理の対象となるデータである小数値を整数値で表現し、前記整数値を畳み込みニューラルネットワークのパラメータとして扱って推論モデルを実行して推論結果を得る推論部と、
- 前記推論結果を入力にして証明生成アルゴリズムを実行して証明を得る証明部と、
- 前記証明を入力にして検証アルゴリズムを実行して検証結果を得る検証部と、
- を備える。

発明の効果

[0009] 本開示によれば、パラメータが小数である推論モデルの検証が可能になる。

図面の簡単な説明

- [0010] [図1]実施の形態1における推論検証システム100の構成図。
[図2]実施の形態1におけるパラメータ生成装置200の構成図。
[図3]実施の形態1における鍵生成装置300の構成図。
[図4]実施の形態1における推論装置400の構成図。
[図5]実施の形態1における証明装置500の構成図。
[図6]実施の形態1における検証装置600の構成図。
[図7]実施の形態1における推論検証方法（パラメータ生成）のフローチャート。
[図8]実施の形態1における推論検証方法（鍵生成）のフローチャート。
[図9]実施の形態1における推論検証方法（推論）のフローチャート。
[図10]実施の形態1における推論検証方法（証明）のフローチャート。
[図11]実施の形態1における推論検証方法（検証）のフローチャート。
[図12]実施の形態1におけるパラメータ生成装置200のハードウェア構成図。
[図13]実施の形態1における鍵生成装置300のハードウェア構成図。
[図14]実施の形態1における推論装置400のハードウェア構成図。
[図15]実施の形態1における証明装置500のハードウェア構成図。
[図16]実施の形態1における検証装置600のハードウェア構成図。

発明を実施するための形態

[0011] 実施の形態および図面において、同じ要素または対応する要素には同じ符号を付している。説明した要素と同じ符号が付された要素の説明は適宜に省略または簡略化する。図中の矢印はデータの流れ又は処理の流れを主に示している。

[0012] 実施の形態1.

推論検証システム100について、図1から図16に基づいて説明する。

[0013] 推論検証システム１００は、推論処理を検証することが可能なシステムである。

[0014] ***構成の説明***

図１に基づいて、推論検証システム１００の構成を説明する。

推論検証システム１００は、パラメータ生成装置２００と鍵生成装置３００と推論装置４００と証明装置５００と検証装置６００といった装置を備える。

これらの装置は、ネットワークを介して互いに通信する。ネットワークの具体例はインターネットである。

[0015] 図２に基づいて、パラメータ生成装置２００の構成を説明する。

パラメータ生成装置２００は、プロセッサ２０１とメモリ２０２と補助記憶装置２０３と通信装置２０４と入出力インタフェース２０５といったハードウェアを備えるコンピュータである。これらのハードウェアは、信号線を介して互いに接続されている。

[0016] プロセッサ２０１は、パラメータ生成装置２００のプロセッサである。

プロセッサは、演算処理を行うＩＣであり、他のハードウェアを制御する。例えば、プロセッサはＣＰＵである。

ＩＣは、Integrated Circuitの略称である。

ＣＰＵは、Central Processing Unitの略称である。

[0017] メモリ２０２は、パラメータ生成装置２００のメモリである。

メモリは、揮発性または不揮発性の記憶装置である。メモリは、主記憶装置またはメインメモリとも呼ばれる。例えば、メモリはＲＡＭである。

メモリ２０２に記憶されたデータは必要に応じて補助記憶装置２０３に保存される。

ＲＡＭは、Random Access Memoryの略称である。

[0018] 補助記憶装置２０３は、パラメータ生成装置２００の補助記憶装置である。

補助記憶装置は不揮発性の記憶装置である。例えば、補助記憶装置は、ROM、HDD、フラッシュメモリまたはこれらの組み合わせである。

補助記憶装置203に記憶されたデータは必要に応じてメモリ202にロードされる。

ROMは、Read Only Memoryの略称である。

HDDは、Hard Disk Driveの略称である。

[0019] 通信装置204は、パラメータ生成装置200の通信装置である。

通信装置はレシーバ及びトランスミッタである。例えば、通信装置は通信チップまたはNICである。

パラメータ生成装置200の通信は通信装置204を用いて行われる。

NICは、Network Interface Cardの略称である。

[0020] 入出力インタフェース205は、パラメータ生成装置200の入出力インタフェースである。

入出力インタフェースは、入力装置および出力装置が接続されるポートである。例えば、入出力インタフェースはUSB端子であり、入力装置はキーボードおよびマウスであり、出力装置はディスプレイである。

パラメータ生成装置200の入出力は入出力インタフェース205を用いて行われる。

USBは、Universal Serial Busの略称である。

[0021] パラメータ生成装置200は、受付部210と生成部220と出力部230といった要素を備える。これらの要素はソフトウェアで実現される。

[0022] 補助記憶装置203には、受付部210と生成部220と出力部230としてコンピュータを機能させるためのパラメータ生成プログラムが記憶されている。パラメータ生成プログラムは、メモリ202にロードされて、プロセッサ201によって実行される。

補助記憶装置203には、さらに、OSが記憶されている。OSの少なくとも一部は、メモリにロードされて、プロセッサによって実行される。

プロセッサ２０１は、ＯＳを実行しながら、パラメータ生成プログラムを実行する。

ＯＳは、Operating Systemの略称である。

[0023] パラメータ生成プログラムの入出力データは記憶部２９０に記憶される。

メモリ２０２は記憶部２９０として機能する。但し、補助記憶装置２０３、プロセッサ２０１内のレジスタおよびプロセッサ２０１内のキャッシュメモリなどの記憶装置が、メモリ２０２の代わりに、又は、メモリ２０２と共に、記憶部２９０として機能してもよい。

[0024] パラメータ生成装置２００は、プロセッサ２０１を代替する複数のプロセッサを備えてもよい。

[0025] 図３に基づいて、鍵生成装置３００の構成を説明する。

鍵生成装置３００は、プロセッサ３０１とメモリ３０２と補助記憶装置３０３と通信装置３０４と入出力インタフェース３０５といったハードウェアを備えるコンピュータである。これらのハードウェアは、信号線を介して互いに接続されている。

[0026] プロセッサ３０１は、鍵生成装置３００のプロセッサである。

メモリ３０２は、鍵生成装置３００のメモリである。

補助記憶装置３０３は、鍵生成装置３００の補助記憶装置である。

通信装置３０４は、鍵生成装置３００の通信装置である。

入出力インタフェース３０５は、鍵生成装置３００の入出力インタフェースである。

[0027] 鍵生成装置３００は、受付部３１０と生成部３２０と出力部３３０といった要素を備える。これらの要素はソフトウェアで実現される。

[0028] 補助記憶装置３０３には、受付部３１０と生成部３２０と出力部３３０としてコンピュータを機能させるための鍵生成プログラムが記憶されている。鍵生成プログラムは、メモリ３０２にロードされて、プロセッサ３０１によって実行される。

鍵生成装置３００には、さらに、ＯＳが記憶されている。

プロセッサ３０１は、ＯＳを実行しながら、鍵生成プログラムを実行する。

[0029] 鍵生成プログラムの入出力データは記憶部３９０に記憶される。

メモリ３０２は記憶部３９０として機能する。但し、補助記憶装置３０３、プロセッサ３０１内のレジスタおよびプロセッサ３０１内のキャッシュメモリなどの記憶装置が、メモリ３０２の代わりに、又は、メモリ３０２と共に、記憶部３９０として機能してもよい。

[0030] 鍵生成装置３００は、プロセッサ３０１を代替する複数のプロセッサを備えてもよい。

[0031] 図４に基づいて、推論装置４００の構成を説明する。

推論装置４００は、プロセッサ４０１とメモリ４０２と補助記憶装置４０３と通信装置４０４と入出力インタフェース４０５といったハードウェアを備えるコンピュータである。これらのハードウェアは、信号線を介して互いに接続されている。

[0032] プロセッサ４０１は、推論装置４００のプロセッサである。

メモリ４０２は、推論装置４００のメモリである。

補助記憶装置４０３は、推論装置４００の補助記憶装置である。

通信装置４０４は、推論装置４００の通信装置である。

入出力インタフェース４０５は、推論装置４００の入出力インタフェースである。

[0033] 推論装置４００は、受付部４１０と推論部４２０と出力部４３０といった要素を備える。これらの要素はソフトウェアで実現される。

[0034] 補助記憶装置４０３には、受付部４１０と推論部４２０と出力部４３０としてコンピュータを機能させるための推論プログラムが記憶されている。推論プログラムは、メモリ４０２にロードされて、プロセッサ４０１によって実行される。

推論装置４００には、さらに、ＯＳが記憶されている。

プロセッサ４０１は、ＯＳを実行しながら、推論プログラムを実行する。

[0035] 推論プログラムの入出力データは記憶部４９０に記憶される。

メモリ４０２は記憶部４９０として機能する。但し、補助記憶装置４０３、プロセッサ４０１内のレジスタおよびプロセッサ４０１内のキャッシュメモリなどの記憶装置が、メモリ４０２の代わりに、又は、メモリ４０２と共に、記憶部４９０として機能してもよい。

[0036] 推論装置４００は、プロセッサ４０１を代替する複数のプロセッサを備えてもよい。

[0037] 図５に基づいて、証明装置５００の構成を説明する。

証明装置５００は、プロセッサ５０１とメモリ５０２と補助記憶装置５０３と通信装置５０４と入出力インタフェース５０５といったハードウェアを備えるコンピュータである。これらのハードウェアは、信号線を介して互いに接続されている。

[0038] プロセッサ５０１は、証明装置５００のプロセッサである。

メモリ５０２は、証明装置５００のメモリである。

補助記憶装置５０３は、証明装置５００の補助記憶装置である。

通信装置５０４は、証明装置５００の通信装置である。

入出力インタフェース５０５は、証明装置５００の入出力インタフェースである。

[0039] 証明装置５００は、受付部５１０と保管部５２０と証明部５３０と出力部５４０といった要素を備える。保管部５２０は、鍵保管部５２１と推論結果保管部５２２を備える。これらの要素はソフトウェアで実現される。

[0040] 補助記憶装置５０３には、受付部５１０と保管部５２０と証明部５３０と出力部５４０としてコンピュータを機能させるための証明プログラムが記憶されている。証明プログラムは、メモリ５０２にロードされて、プロセッサ５０１によって実行される。

証明装置５００には、さらに、ＯＳが記憶されている。

プロセッサ５０１は、ＯＳを実行しながら、証明プログラムを実行する。

[0041] 証明プログラムの入出力データは記憶部５９０に記憶される。

メモリ５０２は記憶部５９０として機能する。但し、補助記憶装置５０３、プロセッサ５０１内のレジスタおよびプロセッサ５０１内のキャッシュメモリなどの記憶装置が、メモリ５０２の代わりに、又は、メモリ５０２と共に、記憶部５９０として機能してもよい。

[0042] 証明装置５００は、プロセッサ５０１を代替する複数のプロセッサを備えてもよい。

[0043] 図６に基づいて、検証装置６００の構成を説明する。

検証装置６００は、プロセッサ６０１とメモリ６０２と補助記憶装置６０３と通信装置６０４と入出力インタフェース６０５といったハードウェアを備えるコンピュータである。これらのハードウェアは、信号線を介して互いに接続されている。

[0044] プロセッサ６０１は、検証装置６００のプロセッサである。

メモリ６０２は、検証装置６００のメモリである。

補助記憶装置６０３は、検証装置６００の補助記憶装置である。

通信装置６０４は、検証装置６００の通信装置である。

入出力インタフェース６０５は、検証装置６００の入出力インタフェースである。

[0045] 検証装置６００は、受付部６１０と保管部６２０と検証部６３０と出力部６４０といった要素を備える。保管部６２０は、鍵保管部６２１と証明保管部６２２を備える。これらの要素はソフトウェアで実現される。

[0046] 補助記憶装置６０３には、受付部６１０と保管部６２０と検証部６３０と出力部６４０としてコンピュータを機能させるための検証プログラムが記憶されている。検証プログラムは、メモリ６０２にロードされて、プロセッサ６０１によって実行される。

検証装置６００には、さらに、ＯＳが記憶されている。

プロセッサ６０１は、ＯＳを実行しながら、検証プログラムを実行する。

[0047] 検証プログラムの入出力データは記憶部６９０に記憶される。

メモリ６０２は記憶部６９０として機能する。但し、補助記憶装置６０３

、プロセッサ601内のレジスタおよびプロセッサ601内のキャッシュメモリなどの記憶装置が、メモリ602の代わりに、又は、メモリ602と共に、記憶部690として機能してもよい。

[0048] 検証装置600は、プロセッサ601を代替する複数のプロセッサを備えてもよい。

[0049] ***記法の説明***

以降の説明における記法を示す。

「A」が分布であるときに、「 $y \leftarrow A$ 」は、yをAからその分布に従ってランダムに選ぶことを指す。

「A」が集合であるときに、「 $y \leftarrow A$ 」は、入力xに対して、yをA"から一様に選ぶことを指す。

「p」は、任意の素数である。

「G」は、位数pの群である。

「m」および「n」は、任意の自然数である。

「e」は、自然対数の底である。

「H」は、ハッシュ関数である。

「D」は、推論モデルの入力データの集合である。

「x」は、推論処理の対象となる入力データである。

「M」は、推論モデルである。

「n」が自然数であるとき、[n]は集合{1, ..., n}である。

「Z」は、整数の集合である。

「R」は、実数の集合である。

[0050] ***動作の説明***

推論検証システム100の動作の手順は推論検証方法に相当する。また、推論検証システム100の動作の手順は推論検証プログラムによる処理の手順に相当する。

推論検証プログラムは、パラメータ生成プログラムと鍵生成プログラムと推論プログラムと証明プログラムと検証プログラムを含む。

各パラメータ生成プログラムは、光ディスクまたはフラッシュメモリ等の不揮発性の記録媒体にコンピュータ読み取り可能に記録（格納）することができる。

[0051] 図7に基づいて、推論検証方法におけるパラメータ生成装置200の動作を説明する。

ステップS210において、受付部210は、パラメータ生成装置200に入力されるパラメータ（ λ , D ）を受け付ける。

例えば、パラメータ（ λ , D ）は、管理者によってパラメータ生成装置200に入力される。

[0052] ステップS220において、生成部220は、パラメータ（ λ , D ）を入力にしてセットアップアルゴリズムを実行する。

これにより、公開パラメータ pp が生成される。

[0053] セットアップアルゴリズムは、公開パラメータ pp を生成するためのアルゴリズムである。

例えば、セットアップアルゴリズム（ $Setup$ ）は、以下のように表される。

[0054] [数1]

$$Setup(1^\lambda, D):$$
$$g_i \leftarrow \mathcal{G}$$
$$pp := (g_1, \dots, g_n)$$

[0055] ステップS230において、出力部230は、公開パラメータ pp を出力する。

具体的には、出力部230は、公開パラメータ pp を鍵生成装置300へ送信する。

[0056] 図8に基づいて、推論検証方法における鍵生成装置300の動作を説明する。

ステップS 3 1 0において、受付部3 1 0は、鍵生成装置3 0 0に入力される公開パラメータ pp を受け付ける。

具体的には、受付部3 1 0は、パラメータ生成装置2 0 0から公開パラメータ pp を受信する。

[0057] ステップS 3 2 0において、生成部3 2 0は、公開パラメータ pp を入力にして鍵生成アルゴリズムを実行する。

これにより、公開鍵 pk と秘密鍵 sk が生成される。

[0058] 鍵生成アルゴリズムは、公開鍵 pk と秘密鍵 sk を生成するためのアルゴリズムである。

例えば、鍵生成アルゴリズムは (Kg) は、以下のように表される。

[0059] [数2]

$Kg(pp):$

$$h_i \leftarrow G$$

$$s_j \leftarrow Z_p$$

$$pk := (g_1, \dots, g_n, h_1, \dots, h_m)$$

$$sk := (s_1, \dots, s_l)$$

[0060] ステップS 3 3 0において、出力部3 3 0は、公開鍵 pk と秘密鍵 sk を出力する。

具体的には、出力部3 3 0は、公開鍵 pk と秘密鍵 sk を証明装置5 0 0へ送信する。また、出力部3 3 0は、公開鍵 pk を検証装置6 0 0へ送信する。

[0061] 図4に基づいて、推論検証方法における推論装置4 0 0の動作を説明する。

ステップS 4 1 0において、受付部4 1 0は、推論装置4 0 0に入力されるパラメータ (M, x) を受け付ける。

例えば、推論モデル M は、推論モデル M の提供者によって推論装置4 0 0

に入力される。また、データ x は、クライアントによって推論装置 400 へ送信される。

推論モデル M は、推論処理のためのモデルである。

データ x は、推論処理の対象となるデータである。

[0062] ステップ S 420 において、推論部 420 は、推論モデル M とデータ x を入力にして推論処理アルゴリズムを実行する。

これにより、推論結果 c が生成される。

[0063] 推論処理アルゴリズムは、推論結果 c を生成するためのアルゴリズムである。

例えば、推論処理アルゴリズム (Classify) は、以下のように表される。

CNN は、推論モデル M によって実行される畳み込みニューラルネットワークである。

[0064] [数3]

$$\text{Classify}(M, x):$$
$$c := \text{CNN}(x)$$

[0065] ステップ S 430 において、出力部 430 は、推論結果 c を出力する。

具体的には、出力部 430 は、推論結果 c を証明装置 500 へ送信する。

[0066] 図 10 に基づいて、推論検証方法における証明装置 500 の動作を説明する。

ステップ S 510 において、受付部 510 は、公開鍵 p_k と秘密鍵 s_k と推論結果 c を受け付ける。

具体的には、受付部 510 は、鍵生成装置 300 から公開鍵 p_k と秘密鍵 s_k を受信する。また、受付部 510 は、推論装置 400 から推論結果 c を受信する。

[0067] ステップ S 520 において、鍵保管部 521 は、公開鍵 p_k と秘密鍵 s_k を保管する。

また、推論結果保管部 522 は、推論結果 c を保管する。

例えば、公開鍵 pk と秘密鍵 sk と推論結果 c は、補助記憶装置 503 に記憶される。

[0068] ステップ S530 において、証明部 530 は、公開鍵 pk と秘密鍵 sk と推論結果 c を入力にして証明生成アルゴリズムを実行する。

これにより、証明 P が生成される。

[0069] 証明生成アルゴリズムは、証明 P を生成するためのアルゴリズムである。

証明生成アルゴリズム (Prove) の例について、以下に説明する。

[0070] 推論結果 c は、次のように表される。 c_i は、CNN の i 層目の計算結果である。

$$c = c_1, \dots, c_n$$

[0071] $Prove_i$ は、計算結果 c_i に対して証明 P_i を生成するためのアルゴリズムである。

例えば、証明生成アルゴリズム (Prove) は、以下のように表される。

[0072] [数4]

$Prove(pk, sk, c):$

$$P_1 := Prove_1(pk, sk, c_1), \dots, P_n := Prove_n(pk, sk, c_n)$$

$$P := (P_1, \dots, P_n)$$

[0073] $Prove_i$ の種類 ($P_a \sim P_f$) は、以下の通りである。

(P_a) CNN の第 i 層が ReLU 活性化層である場合、 $Prove_i$ の種類は $Prove_{ReLU}$ である。

(P_b) CNN の第 i 層が Affine 層である場合、 $Prove_i$ の種類は $Prove_{Affine}$ である。

(P_c) CNN の第 i 層が畳み込み層である場合、 $Prove_i$ の種類は $Prove_{Conv}$ である。

(P_d) CNNの第 i 層がAverage Pooling層である場合、 $Prove_i$ の種類は $Prove_{AP}$ である。

(P_e) CNNの第 i 層がMax Pooling層である場合、 $Prove_i$ の種類は $Prove_{MP}$ である。

(P_f) CNNの第 i 層がSoft Max層である場合、 $Prove_i$ の種類は $Prove_{SoftMax}$ である。

[0074] $Prove_i$ の各種類 ($P_a \sim P_f$) について後述する。

[0075] ステップS540において、出力部540は、証明Pを出力する。

具体的には、出力部540は、証明Pを検証装置600へ送信する。

[0076] 図11に基づいて、推論検証方法における検証装置600の動作を説明する。

ステップS610において、受付部610は、公開鍵pkと証明Pを受け付ける。

具体的には、受付部610は、鍵生成装置300から公開鍵pkを受信する。また、受付部610は、証明装置500から証明Pを受信する。

[0077] ステップS620において、鍵保管部621は、公開鍵pkを保管する。

また、証明保管部622は、証明Pを保管する。

例えば、公開鍵pkと証明Pは、補助記憶装置603に記憶される。

[0078] ステップS630において、検証部630は、公開鍵pkと証明Pを入力にして検証アルゴリズムを実行する。

これにより、検証結果Vが生成される。

[0079] 検証アルゴリズムは、検証結果Vを生成するためのアルゴリズムである。

検証アルゴリズム (Verify) の例について、以下に説明する。

[0080] $Verify_i$ は、証明 P_i を検証するためのアルゴリズムである。

例えば、検証アルゴリズム (Verify) は、以下のように表される。

[0081]

[数5]

Verify(pk, P):

$$V_1 := \text{Verify}_1(pk, P_1), \dots, \text{Verify}_n(pk, P_n)$$

$$\text{if}(V_1 \wedge \dots \wedge V_n = \text{true})$$

$$V := c$$

$$\text{if}(V_1 \wedge \dots \wedge V_n = \text{false})$$

$$V := \perp$$

[0082] Verify_i の種類 ($V_a \sim V_f$) は、以下の通りである。

(V_a) CNNの第*i*層がReLU活性化層である場合、 Verify_i の種類は $\text{Verify}_{\text{ReLU}}$ である。

(V_b) CNNの第*i*層がAffine層である場合、 Verify_i の種類は $\text{Verify}_{\text{Affine}}$ である。

(V_c) CNNの第*i*層が畳み込み層である場合、 Verify_i の種類は $\text{Verify}_{\text{Conv}}$ である。

(V_d) CNNの第*i*層がAverage Pooling層である場合、 Verify_i の種類は $\text{Verify}_{\text{AP}}$ である。

(V_e) CNNの第*i*層がMax Pooling層である場合、 Verify_i の種類は $\text{Verify}_{\text{MP}}$ である。

(V_f) CNNの第*i*層がSoftMax層である場合、 Verify_i の種類は $\text{Verify}_{\text{SoftMax}}$ である。

[0083] Verify_i の各種類 ($V_a \sim V_f$) について後述する。

[0084] ステップS640において、出力部640は、検証結果Vを出力する。

例えば、出力部640は、検証結果Vをディスプレイに表示する。

[0085] ***プロトコルおよび整数値表示の説明***

実施の形態1で使用されるゼロ知識証明のプロトコルに関して、以下のプ

ロトコルを説明する。また、実施の形態1における小数の整数値表示を説明する。

- (1) Schnorrプロトコル
- (2) 複数指数に対するSchnorrプロトコル
- (3) 一般化Schnorrのプロトコル
- (4) OR Proofプロトコル
- (5) nOR Proofプロトコル
- (6) 小数の整数値表示
- (7) Range Proofsプロトコル
- (8) Multiplication Proofsプロトコル
- (9) The ReLU Layerプロトコル [P_a , V_a]
- (10) The Affine Layerプロトコル [P_b , V_b]
- (11) The Convolution Layerプロトコル [P_c , V_c]
- (12) The Average Pooling Layerプロトコル [P_d , V_d]
- (13) The Max Pooling Layerプロトコル [P_e , V_e]
- (14) The SoftMax Layerプロトコル [P_f , V_f]

[0086] (1) Schnorrプロトコルについて説明する。

「 g 」が G の生成元である。このとき、 $x \in \mathbb{Z}_p$ に対して $y = g^x$ が成り立つ。

Schnorrプロトコルは、検証者に対して x に関する情報を一切与えずに、証明者が $y = g^x$ における x を知っていることを証明するためのプロトコルである。

Schnorrプロトコルは、証明生成アルゴリズム $\text{Prove}_{\text{Schnorr}}$ と、検証アルゴリズム $\text{Verify}_{\text{Schnorr}}$ と、で構成される。

[0087] $\text{Prove}_{\text{Schnorr}}$ は、証明 P を出力する。

[0088] [数11]

$$\text{Prove}_{\text{Schnorr}}(g, y, x):$$

$$r \leftarrow Z_p$$

$$R := g^r$$

$$C := H(g, y, R)$$

$$s := r + Cx$$

$$P = (g, y, C, s)$$

[0089] $\text{Verify}_{\text{Schnorr}}$ は、trueまたはfalseを出力する。

[0090] [数12]

$$\text{Verify}_{\text{Schnorr}}(g, y, C, s):$$

$$HV := H(g, y, g^s/y^C)$$

$$\text{if}(C = HV) \quad \text{true}$$

$$\text{if}(C \neq HV) \quad \text{false}$$

[0091] (2) 複数指数に対するSchnorrプロトコルについて説明する。複数指数に対するSchnorrプロトコルは、(1) Schnorrプロトコルを一般化して得られる。

$g_1, \dots, g_n$ がGの生成元である。このとき、 $x_1 \in Z_p, \dots, x_n \in Z_p$ に対して式(2A)が成り立つ。

[0092] [数13]

$$y = \prod_{i=1}^n g_i^{x_i} \quad (2A)$$

[0093] 複数指数に対するSchnorrプロトコルは、検証者に対して($x_1, \dots, x_n$)に関する情報を一切与えずに、証明者が式(2A)における(x

$x_1, \dots, x_n$) を知っていることを証明するためのプロトコルである。

複数指数に対する Schnorr プロトコルは、証明作成アルゴリズム $\text{Prove}_{\text{MultiSchnorr}}$ と、検証アルゴリズム $\text{Verify}_{\text{MultiSchnorr}}$ と、で構成される。

[0094] $\text{Prove}_{\text{MultiSchnorr}}$ は、証明 P を出力する。

[0095] [数14]

$\text{Prove}_{\text{MultiSchnorr}}(g_1, \dots, g_n, y, x_1, \dots, x_n):$

$$r_i \leftarrow Z_p \quad (i \in \{0, \dots, n\})$$

$$R := \prod_{i=1}^n g_i^{r_i}$$

$$C := H(g_1, \dots, g_n, y, R)$$

$$s_i := r_i + Cx_i \quad (i \in \{0, \dots, n\})$$

$$P = (g_1, \dots, g_n, y, C, s_1, \dots, s_n)$$

[0096] $\text{Verify}_{\text{MultiSchnorr}}$ は、 true または false を出力する。

[0097] [数15]

$\text{Verify}_{\text{MultiSchnorr}}(g_1, \dots, g_n, y, C, s_1, \dots, s_n):$

$$HV := H\left(g_1, \dots, g_n, y, \prod_{i=1}^n g_i^{s_i} / y^C\right)$$

$$\text{if}(C = HV) \quad \text{true}$$

$$\text{if}(C \neq HV) \quad \text{false}$$

[0098] (3) 一般化 Schnorr プロトコルについて説明する。一般化 Schnorr プロトコルは、(2) 複数指数に対する Schnorr プロトコルを一般化して得られる。

$g_{1,1}, \dots, g_{1,n}, g_{2,1}, \dots, g_{m,n}$ が G の mn 個の生成元である。このとき、 $x_1 \in Z_p, \dots, x_n \in Z_p$ に対して式 (3A) が成り立

つ。

[0099] [数16]

$$\bigwedge_{j=1}^m \left(\prod_{i=1}^n g_{j,i}^{x_i} = y_j \right) \quad (3A)$$

[0100] 一般化Schnorrプロトコルは、検証者に対して $(x_1, \dots, x_n)$ に関する情報を一切与えずに、証明者が式(3A)における $(x_1, \dots, x_n)$ を知っていることを証明するためのプロトコルである。

一般化Schnorrプロトコルは、証明作成アルゴリズム $\text{Prove}_{\text{GenSchnorr}}$ と、検証アルゴリズム $\text{Verify}_{\text{GenSchnorr}}$ と、で構成される。

[0101] $\text{Prove}_{\text{GenSchnorr}}$ は、証明 P を出力する。

[0102] [数17]

$$\begin{aligned} & \text{Prove}_{\text{GenSchnorr}} \left((g_{j,i})_{j \in [m], i \in [n]}, (y_j)_{j \in [m]}, (x_i)_{i \in [n]} \right) : \\ & \quad r_1, \dots, r_n \leftarrow Z_p \\ & \quad R_1 := \prod_{i=1}^n g_{1,i}^{r_i}, \dots, R_m := \prod_{i=1}^n g_{m,i}^{r_i} \\ & \quad C := H \left((g_{j,i})_{j \in [m], i \in [n]}, (y_j)_{j \in [m]}, (R_j)_{j \in [m]} \right) \\ & \quad s_1 := r_1 + Cx_1, \dots, s_n := r_n + Cx_n \\ & \quad P := \left((g_{j,i})_{j \in [m], i \in [n]}, (y_j)_{j \in [m]}, C, s_1, \dots, s_n \right) \end{aligned}$$

[0103] $\text{Verify}_{\text{GenSchnorr}}$ は、trueまたはfalseを出力する。

[0104]

[数18]

$$\text{Verify}_{\text{GenSchnorr}} \left((g_{j,i})_{j \in [m], i \in [n]}, (y_j)_{j \in [m]}, C, s_1, \dots, s_n \right) :$$

$$HV := H \left((g_{j,i})_{j \in [m], i \in [n]}, (y_j)_{j \in [m]}, \left(\prod_{i=1}^n g_{j,i}^{s_i} / y_j^C \right)_{j \in [m]} \right)$$

$$\text{if}(C = HV) \quad \text{true}$$

$$\text{if}(C \neq HV) \quad \text{false}$$

[0105] (4) OR Proof プロトコルについて説明する。

$g_{1,1}, \dots, g_{1,n}, g_{2,1}, \dots, g_{m,n}$ が G の mn 個の生成元である。このとき、 $x = (x_1, \dots, x_n)$ に対して式 (4A) が成り立つ。

[0106] [数19]

$$f(x) = \left(\prod_{i=1}^n g_{j,i}^{x_i} \right)_{j=1,\dots,m} \quad (4A)$$

[0107] また、 $i = 1, 2$ に対して以下の式が成り立つ。

$$x^{(i)} = (x_1^{(i)}, \dots, x_n^{(i)}) \in (Z_p)^n$$

$$y^{(i)} = f(x^{(i)})$$

[0108] OR Proof プロトコルは、検証者に対して $x^{(1)}$ と $x^{(2)}$ に関する情報を一切与えずに、証明者が $y^{(1)} = f(x^{(1)})$ における $x^{(1)}$ 、または、 $y^{(2)} = f(x^{(2)})$ における $x^{(2)}$ を知っていることを証明するためのプロトコルである。

OR Proof のプロトコルは、証明作成アルゴリズム Prove_{OR} と、検証アルゴリズム $\text{Verify}_{\text{OR}}$ と、で構成される。

[0109] Prove_{OR} は、証明 P を出力する。

[0110]

[数20]

$$\begin{aligned}
& \text{Prove}_{\text{OR}} \left((g_{j,i})_{j \in [m], i \in [n]}, (y^{(i)}, x^{(i)})_{i=1,2} \right): \\
& \quad j := 3 - i \quad (i = 1 \text{ or } 2) \\
& \quad s^{(j)} \leftarrow (Z_p)^n \\
& \quad C^{(j)} \leftarrow Z_p \\
& \quad R^{(j)} := f(s^{(j)}) / (y^{(j)})^{c^{(j)}} \\
& \quad r^{(i)} \leftarrow (Z_p)^n \\
& \quad R^{(i)} := f(r^{(i)}) \\
& \quad C := H(y^{(1)}, y^{(2)}, R^{(1)}, R^{(2)}) \\
& \quad C^{(i)} := C - C^{(j)} \\
& \quad s^{(i)} := r^{(i)} + C^{(i)} x^{(i)} \\
& \quad P := \left((g_{j,i})_{j \in [m], i \in [n]}, y^{(1)}, y^{(2)}, C^{(1)}, C^{(2)}, s^{(1)}, s^{(2)} \right)
\end{aligned}$$

[0111] $\text{Verify}_{\text{OR}}$ は、 true または false を出力する。

[0112] [数21]

$$\begin{aligned}
& \text{Verify}_{\text{OR}} \left((g_{j,i})_{j \in [m], i \in [n]}, y^{(1)}, y^{(2)}, C^{(1)}, C^{(2)}, s^{(1)}, s^{(2)} \right): \\
& \quad HV := H \left((g_{j,i})_{j \in [m], i \in [n]} y^{(1)}, y^{(2)}, f(s^{(1)}) \right. \\
& \quad \quad \left. / (y^{(1)})^{c^{(1)}}, f(s^{(2)}) / (y^{(2)})^{c^{(2)}} \right) \\
& \quad \text{if}(C^{(1)} + C^{(2)} = HV) \quad \text{true} \\
& \quad \text{if}(C^{(1)} + C^{(2)} \neq HV) \quad \text{false}
\end{aligned}$$

[0113] (5) nOR Proof プロトコルについて説明する。 nOR Pro

o f プロトコルは、(4) O R P r o o f プロトコルを一般化して得られる。

$g_{1,1}, \dots, g_{1,n}, g_{2,1}, \dots, g_{m,n}$ が G の mn 個の生成元である。このとき、 $x = (x_1, \dots, x_n)$ に対して式 (5A) が成り立つ。

[0114] [数22]

$$f(x) = \left(\prod_{i=1}^n g_{j,i}^{x_i} \right)_{j=1,\dots,m} \quad (5A)$$

[0115] また、 $i \in [n]$ に対して以下の式が成り立つ。

$$x^{(i)} = (x_1^{(i)}, \dots, x_n^{(i)}) \in (Z_p)^n$$

$$y^{(i)} = f(x^{(i)})$$

[0116] n O R P r o o f プロトコルは、検証者に対して各 $x^{(i)}$ に関する情報を一切与えずに、証明者が $y^{(i)} = f(x^{(i)})$ における $x^{(i)}$ のうちの少なくとも1つは知っていることを証明するためのプロトコルである。

n O R P r o o f プロトコルは、証明作成アルゴリズム $P r o v e n_{OR}$ と、検証アルゴリズム $V e r i f y n_{OR}$ と、で構成される。

$P r o v e n_{OR}$ は、全ての $j \in [n] \setminus \{i\}$ に対して $s(j)$ と $C(j)$ と $R(j)$ の3つの式を計算し、証明 P を出力する。

[0117]

[数23]

$$\begin{aligned}
& \text{Prove}_{\text{nOR}} \left((g_{j,i})_{j \in [m], i \in [n]}, (y^{(i)}, x^{(i)})_{i \in [n]} \right) : \\
& \quad s^{(j)} \leftarrow (Z_p)^n \\
& \quad C^{(j)} \leftarrow Z_p \\
& \quad R^{(j)} := f(s^{(j)}) / (y^{(j)})^{c^{(j)}} \\
& \quad r^{(i)} \leftarrow (Z_p)^n \\
& \quad R^{(i)} := f(r^{(i)}) \\
& \quad C := H \left((y^{(i)}, R^{(i)})_{i \in [n]} \right) \\
& \quad C^{(i)} := C - \sum_{j \in [n] \setminus \{i\}} C^{(j)} \\
& \quad s^{(i)} := r^{(i)} + C^{(i)} x^{(i)} \\
& \quad P := \left((g_{j,i})_{j \in [m], i \in [n]}, (y^{(i)}, C^{(i)}, s^{(i)})_{i \in [n]} \right)
\end{aligned}$$

[0118] $\text{Verify}_{\text{nOR}}$ は、 true または false を出力する。

[0119] [数24]

$$\begin{aligned}
& \text{Verify}_{\text{nOR}} \left((g_{j,i})_{j \in [m], i \in [n]}, (y^{(i)}, C^{(i)}, s^{(i)})_{i \in [n]} \right) : \\
& \quad HV := H \left(\left(y^{(i)}, f(s^{(i)}) / (y^{(i)})^{c^{(i)}} \right)_{i \in [n]} \right) \\
& \quad \text{if}(C^{(1)} + \dots + C^{(n)} = HV) \quad \text{true} \\
& \quad \text{if}(C^{(1)} + \dots + C^{(n)} \neq HV) \quad \text{false}
\end{aligned}$$

[0120] (6) 小数の整数値表示について説明する。

「l」, 「d」は正の整数であり、式(6A)が成り立つ。

[0121] [数25]

$$2^{2(l+d)+1} + 2^d \leq (p-1)/2 \quad (6A)$$

[0122] このとき、小数 x の固定点表示は整数 $\langle x \rangle$ で表現される。

[0123] [数26]

$$x = x_0 + x_1 2^{-d} \quad (x_0 \in \{-2^l, \dots, 2^l - 1\}, x_1 \in \{0, \dots, 2^d - 1\})$$

$$\langle x \rangle = x_0 2^d + x_1 \in Z_p$$

[0124] これにより、通常は実数で表現される畳み込みニューラルネットワークのモデルパラメータを、整数値として扱うことが可能になる。

[0125] (7) Range Proofs プロトコルについて説明する。

「 g 」, 「 h 」は、 G の生成元である。

「 t 」は、整数である。

Range Proofs プロトコルは、整数 t に関する情報を検証者に与えることなく、整数 t が 0 以上 2^m 未満の整数であることを証明者が証明するためのプロトコルである。

Range Proofs プロトコルは、証明作成アルゴリズム $\text{Prove}_{\text{Range}}$ と、検証アルゴリズム $\text{Verify}_{\text{Range}}$ と、で構成される。

[0126] $\text{Prove}_{\text{Range}}$ は、 t の 2 進数表示の式を計算する。

[0127] [数27]

$$t = \sum_{i=1}^{m-1} 2^i t_i \quad (t_i \in \{0, 1\})$$

$$s_1 \leftarrow Z_p, \dots, s_{m-1} \leftarrow Z_p$$

$$B_1 := g^{s_1} h^{t_1}, \dots, B_{m-1} := g^{s_{m-1}} h^{t_{m-1}}$$

[0128] そして、 $\text{Prove}_{\text{Range}}$ は、(3) 一般化 Schnorr プロトコルにお

ける $\text{Prove}_{\text{GenSchnorr}}$ を用いて、以下の4つの式に対する証明 P を算出する。

[0129] [数28]

$\text{Prove}_{\text{Range}}(g, h, t):$

$$A = g^s h^t$$

$$B_i = g^{s_i} h^{t_i} \quad (i \in \{0, \dots, m-1\})$$

$$B_i^{t_i} / B_i = g^{s_i t_i - s_i} \quad (i \in \{0, \dots, m-1\})$$

$$A = g^s \prod_{i=1}^{m-1} h^{2^i t_i}$$

[0130] $\text{Verify}_{\text{Range}}$ は、検証結果 V を出力する。

[0131] [数29]

$\text{Verify}_{\text{Range}}(P):$

$$V := \text{Verify}_{\text{GenSchnorr}}(P)$$

[0132] (8) $\text{Multiplication Proofs}$ プロトコルについて説明する。

「 g 」, 「 h 」は、 G の生成元である。このとき、整数 x , y , z に対して以下の式が成り立つ。

$$\langle x \rangle = x_0 + x_1 2^d$$

$$\langle y \rangle = y_0 + y_1 2^d$$

$$\langle z \rangle = z_0 + z_1 2^d$$

$\text{Multiplication Proofs}$ プロトコルは、 z , x , y に関する情報を検証者に与えることなく、整数 x , y , z に対して $\langle z \rangle = \langle x y \rangle$ の関係が切り捨てを除いて成立することを証明者が証明するためのプロトコルである。

M u l t i p l i c a t i o n P r o o f s プロトコルは、証明作成アルゴリズム $P r o v e_{M u l t}$ と、検証アルゴリズム $V e r i f y_{M u l t}$ と、で構成される。

[0133] $P r o v e_{M u l t}$ において、各符号の意味は以下の通りである。

[0134] [数30]

$$r_x, r_y, r_z \leftarrow Z_p$$

$$w := x_1 y_1 \bmod 2^d$$

$$r_w \leftarrow Z_p$$

$$C_w := g^{r_w} h^w$$

[0135] $P r o v e_{M u l t}$ は、(3) 一般化 S c h n o r r プロトコルと (7) R a n g e P r o o f s プロトコルを用いて、以下の9つの式に対する証明 P を算出する。

[0136]

[数31]

$$\text{Prove}_{\text{Mult}}(g, h, x, y, z):$$

$$C_x = g^{r_x} h^{\langle x \rangle}$$

$$C_y = g^{r_y} h^{\langle y \rangle}$$

$$\langle x \rangle \in \{-2^{l+d}, \dots, 2^{l+d} - 1\}$$

$$\langle y \rangle \in \{-2^{l+d}, \dots, 2^{l+d} - 1\}$$

$$C_z = g^{r_z} h^{\langle z \rangle}$$

$$\langle z \rangle \in \{-2^{l+d}, \dots, 2^{l+d} - 1\}$$

$$C_w = g^{r_w} h^{\langle w \rangle}$$

$$\langle w \rangle \in \{-2^{l+d}, \dots, 2^{l+d} - 1\}$$

$$C_x^{\langle y \rangle} / (C_z^{2^d} h^w) = g^{r_x \langle y \rangle - r_z 2^d}$$

[0137] $\text{Verify}_{\text{Mult}}$ は、検証結果Vを出力する。

[0138] [数32]

$$\text{Verify}_{\text{Mult}}(P):$$

$$V := \text{Verify}_{\text{GenSchnorr}}(P)$$

[0139] (9) The ReLU Rayer プロトコルについて説明する。

ReLU関数は、畳み込みニューラルネットワークの活性化層で用いられる関数である。

[0140] [数33]

$$\text{ReLU}(x) = \begin{cases} 0 & (x < 0) \\ x & (0 \leq x) \end{cases}$$

[0141] The ReLU Rayer プロトコルは、整数 x 、 y に関する情報を検証者に与えることなく、整数 x 、 y に対して $y = \text{ReLU}(x)$ の関係が成立することを証明者が証明するためのプロトコルである。

[0142] The ReLU Rayer プロトコルは、 (P_a) 証明生成アルゴリズム $\text{Prove}_{\text{ReLU}}$ と、 (V_a) 検証アルゴリズム $\text{Verify}_{\text{ReLU}}$ と、で構成される。

[0143] $\text{Prove}_{\text{ReLU}}$ において、各符号の意味は以下の通りである。

[0144] [数34]

$$a := 0$$

$$r_x, r_y, r_a, r_{\text{neg}} \leftarrow \mathbb{Z}_p$$

$$C_{\text{neg}} := g^{r_{\text{neg}}} h^{\langle ax \rangle}$$

[0145] $\text{Prove}_{\text{ReLU}}$ は、(4) OR proof プロトコルと (7) Range Proof プロトコルと (8) Multiplication Proofs プロトコルを用いて、以下の式に対する証明 P を算出する。

[0146] [数35]

$\text{Prove}_{\text{ReLU}}(g, h, x, y):$

$$C_x = g^{r_x} h^{\langle x \rangle}$$

$$\langle x \rangle \in \{-2^{l+d}, \dots, 2^{l+d} - 1\}$$

$$C_y = g^{r_y} h^{\langle y \rangle}$$

$$\langle y \rangle \in \{-2^{l+d}, \dots, 2^{l+d} - 1\}$$

$$C_a = g^{r_a} h^{\langle a \rangle}$$

$$\langle a \rangle \in \{-2^{l+d}, \dots, 2^{l+d} - 1\}$$

$$\langle ax \rangle = \langle a \rangle \langle x \rangle$$

[0147] 証明Pは、次の2つの式のどちらかが成立することを証明する。

[0148] [数36]

$$\langle x \rangle \in \{-2^{l+d}, \dots, 2^{l+d} - 1\} \wedge C_{neg}/C_y = g^\delta$$

$$\langle x \rangle \in \{0, \dots, 2^{l+d} - 1\} \wedge C_x/C_y = g^\delta$$

[0149] $\text{Verify}_{\text{ReLU}}$ は、検証結果Vを出力する。

[0150] [数37]

$\text{Verify}_{\text{ReLU}}(P)$:

$$V := \text{Verify}_{\text{OR}}(P)$$

[0151] (10) The Affine Layerプロトコルについて説明する

。

The Affine Layerプロトコルは、 x 、 y 、 A 、 b に関する情報を検証者に与えることなく、 $A \in \mathbb{R}^{n \times n}$ 、 $b \in \mathbb{R}^n$ 、 $x, y \in \mathbb{R}^n$ に対して $y = Ax + b$ が成り立つことを証明者が証明するためのプロトコルである

。

A 、 b 、 x 、 y は、以下の通りである。

[0152] [数38]

$$A = \begin{pmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{n,1} & \cdots & a_{n,n} \end{pmatrix},$$

$$b = (b_1, \dots, b_n), x = (x_1, \dots, x_n), y = (y_1, \dots, y_n)$$

[0153] The Affine Layer Protocolは、 (P_b) 証明生成アルゴリズム $\text{Prove}_{\text{Affine}}$ と、 (V_b) 検証アルゴリズム $\text{Verify}_{\text{Affine}}$ と、で構成される。

[0154] $\text{Prove}_{\text{Affine}}$ において、各符号の意味は以下の通りである。

[0155] [数39]

$$r_{a_{i,j}}, r_{x_i}, r_{y_i}, r_{i,j}, r_{b_i} \leftarrow Z_p \quad ((i, j) \in [n] \times [n])$$

$$C'_{i,j} := g^{r_{i,j} h^{\langle a_{i,j} x_j \rangle}} \quad ((i, j) \in [n] \times [n])$$

[0156] $\text{Prove}_{\text{Affine}}$ は、(3)一般化Shnorrプロトコルと(7)Range Proofsプロトコルと(8)Multiplication Proofsプロトコルを用いて、以下の式に対する証明Pを算出する。

[0157] [数40]

$\text{Prove}_{\text{Affine}}(g, h, x, y, b, A):$

$$C_{a_{i,j}} = g^{r_{a_{i,j}} h^{\langle a_{i,j} \rangle}} \quad ((i, j) \in [n] \times [n]),$$

$$\langle a_{i,j} \rangle \in \{-2^{l+d}, \dots, 2^{l+d} - 1\} \quad ((i, j) \in [n] \times [n]),$$

$$C_{b_i} = g^{r_{b_i} h^{\langle b_i \rangle}} \quad (i \in [n]),$$

$$\langle b_i \rangle \in \{-2^{l+d}, \dots, 2^{l+d} - 1\} \quad (i \in [n]),$$

$$C_{x_i} = g^{r_{x_i} h^{\langle x_i \rangle}} \quad (i \in [n]),$$

$$\langle x_i \rangle \in \{-2^{l+d}, \dots, 2^{l+d} - 1\} \quad (i \in [n]),$$

$$C_{y_i} = g^{r_{y_i} h^{\langle y_i \rangle}} \quad (i \in [n]),$$

$$\langle y_i \rangle \in \{-2^{l+d}, \dots, 2^{l+d} - 1\} \quad (i \in [n]),$$

$$\langle a_{i,j} x_i \rangle = \langle a_{i,j} \rangle \langle x_i \rangle \quad ((i, j) \in [n] \times [n]),$$

$$C_{y_i} / \left(\prod_{j=1}^n C'_{i,j} \right) C_{b_i} = g^{\delta_i} \quad (i \in [n]).$$

[0158] $\text{Verify}_{\text{Affine}}$ は、検証結果Vを出力する。

[0159]

[数41]

$\text{Verify}_{\text{ReLU}}(P):$

$V := \text{Verify}_{\text{GenShnorr}}(P)$

[0160] (11) The Convolution Layer プロトコルについて説明する。

The Convolution Layer プロトコルは、 x 、 y 、 Conv に関する情報を検証者に与えることなく、 $x, y \in \mathbb{R}^{m \times m'}$ に対して $y = \text{Conv}(x)$ が成り立つことを証明者が証明するためのプロトコルである。

Conv は、畳み込みニューラルネットワークの畳み込み層で入力データ x に対して実行される演算である。

The Convolution Layer プロトコルは、 (P_c) 証明生成アルゴリズム $\text{Prove}_{\text{Conv}}$ と、 (V_c) 検証アルゴリズム $\text{Verify}_{\text{Conv}}$ と、で構成される。

$(a_{i,j})$ は Conv の重みパラメータである。このとき、入力 $x = (x_{1,1}, \dots, x_{1,m'}, \dots, x_{m,1}, \dots, x_{m,m'})$ と出力 $y = (y_{1,1}, \dots, y_{1,m'}, \dots, y_{m,1}, \dots, y_{m,m'})$ に対して、 $y_{i,j}$ は以下の式で表せる。

[0161] [数42]

$$y_{i,j} = \sum_{i'=0}^{s-1} \sum_{j'=0}^{t-1} a_{i,j} x_{i+i',j+j'}$$

$$(1 \leq i \leq m - s + 1, 1 \leq j \leq m' - t + 1)$$

[0162] そのため、(10) The Affine Layer プロトコルと同様の方法で、証明 P および検証結果 V を生成することができる。

[0163]

[数43]

$$\text{Prove}_{\text{Conv}}(g, h, x, y, b, A):$$

$$P := \text{Prove}_{\text{Affinev}}(g, h, x, y, b, A)$$

$$\text{Verify}_{\text{Conv}}(P):$$

$$V := \text{Verify}_{\text{Affine}}(P)$$

[0164] (12) The Average Pooling Layer プロトコルについて説明する。

The Average Pooling Layer プロトコルは、 x 、 y 、 AP に関する情報を検証者に与えることなく、 $x, y \in \mathbb{R}^{m \times m}$ に対して $y = AP(x)$ が成り立つことを証明者が証明するためのプロトコルである。

AP は、畳み込みニューラルネットワークのAverage Pooling層で入力データ x に対して実行される演算である。

The Average Pooling Layer プロトコルは、 (P_d) 証明生成アルゴリズム Prove_{AP} と、 (V_d) 検証アルゴリズム Verify_{AP} と、で構成される。

y, x に対して、以下の関係が成立する。

「 l 」は、 AP のフィルタにおいて、行と列のサイズであり、ストライドである。

[0165] [数44]

$$x = \begin{pmatrix} x_{1,1} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,m} \end{pmatrix}, y = \begin{pmatrix} y_{1,1} & \cdots & y_{1,m} \\ \vdots & \ddots & \vdots \\ y_{m,1} & \cdots & y_{m,m} \end{pmatrix}$$

$$y_{i,j} = \sum_{s=0}^{l-1} \sum_{t=0}^{l-1} \frac{1}{l^2} x_{(l_i+s-1),(l_j+t-1)}$$

$$(1 \leq i \leq \sqrt{m}, 1 \leq j \leq \sqrt{m}).$$

[0166] よって、 y は x の線形変換の形で表せる。

そのため、(10) The Affine Layer プロトコルと同様の方法で、証明 P および検証結果 V を生成できる。

[0167] [数45]

$$\text{Prove}_{\text{AP}}(g, h, x, y, l):$$

$$P := \text{Prove}_{\text{Affinev}}(g, h, x, y, l)$$

$$\text{Verify}_{\text{AP}}(P):$$

$$V := \text{Verify}_{\text{Affine}}(P)$$

[0168] (13) The Max Pooling Layer プロトコルについて説明する。

The Max Pooling Layer プロトコルは、 x 、 y 、 MP に関する情報を検証者に与えることなく、 $x \in \mathbb{R}^{k_m \times k_m}$ 、 $y \in \mathbb{R}^{m \times m}$ に対して $y = MP(x)$ が成り立つことを証明者が証明するためのプロトコルである。

MP は、畳み込みニューラルネットワークの Max Pooling 層で入力データ x に対して実行される演算である。

The Max Pooling Layer プロトコルは、 (P_e) 証明生成アルゴリズム Prove_{MP} と、 (V_e) 検証アルゴリズム Verify_{MP} と、で構成される。

y 、 x に対して、以下の関係が成立する。

「 k 」は、 MP のフィルタにおいて、行と列のサイズであり、ストライドである。

[0169]

[数46]

$$x = \begin{pmatrix} x_{1,1} & \cdots & x_{1,km} \\ \vdots & \ddots & \vdots \\ x_{km,1} & \cdots & x_{km,km} \end{pmatrix}, y = \begin{pmatrix} y_{1,1} & \cdots & y_{1,m} \\ \vdots & \ddots & \vdots \\ y_{m,1} & \cdots & y_{m,m} \end{pmatrix}$$

$$y_{i,j} = \max\{x_{k(i-1)+s,k(j-1)+t} \mid s, t \in [1, k]\}.$$

[0170] この関係が成立することを証明するために、次の式が成立することを証明する。

[0171] [数47]

$$\bigwedge_{s=0}^k \bigwedge_{t=0}^k (y_{i,j} \geq x_{k(i-1)+s,k(j-1)+t})$$

$$\wedge \bigvee_{s=0}^k \bigvee_{t=0}^k (y_{i,j} = x_{k(i-1)+s,k(j-1)+t}).$$

[0172] Prove_{MP} は、全ての $(i, j) \in [m] \times [m]$ に対して、以下の計算を行う。

[0173] [数48]

$\text{Prove}_{\text{MP}}(g, h, x, y, k):$

$$r_{k(i-1)+s,k(j-1)+t} \leftarrow Z_p \quad (s, t) \in [k] \times [k]$$

$$r'_{i,j} \leftarrow Z_p$$

$$C_{k(i-1)+s,k(j-1)+t} := g^{r_{k(i-1)+s,k(j-1)+t}} h^{\langle x_{k(i-1)+s,k(j-1)+t} \rangle}$$

$$(s, t) \in [k] \times [k]$$

$$C'_{i,j} := g^{r'_{i,j}} h^{\langle y_{i,j} \rangle}$$

[0174] Prove_{MP} は、(7) Range Proofs プロトコルと nOR

P r o o f プロトコルを用いて、以下の式に対する証明 P を算出する。

[0175] [数49]

$$\langle y_{i,j} \rangle - \langle x_{k(i-1)+s,k(j-1)+t} \rangle \in \{0, \dots, 2^{l+d+1} - 1\} \quad (s, t) \\ \in [k] \times [k]$$

$$\bigvee_{(s,t) \in [k] \times [k]} C_{k(i-1)+s,k(j-1)+t} / C'_{i,j} \\ C'_{i,j} = \delta_{s,t} \quad (\delta_{s,t} = r_{k(i-1)+s,k(j-1)+t} - r'_{i,j})$$

[0176] V e r i f y_{MP} は、V e r i f y_{nOR} を用いて、検証結果 V を生成する。

[0177] [数50]

Verify_{MP}(P):

$$V := \text{Verify}_{\text{nOR}}(P)$$

[0178] (14) T h e S o f t M a x L a y e r プロトコルについて説明する。

T h e S o f t M a x L a y e r プロトコルは、x、y に関する情報を検証者に与えることなく、 $x, y \in \mathbb{R}^n$ に対して $y = \text{SoftMax}(x)$ が成り立つことを証明者が証明するためのプロトコルである。

S o f t M a x は、畳み込みニューラルネットワークのソフトマックス層で入力データ x に対して実行される演算である。

$x = (x_1, \dots, x_n)$ 、 $y = (y_1, \dots, y_n)$ に対して、 y_i は式 (13A) で表される。

[0179] [数51]

$$y_i = e^{x_i} / \sum_{j=1}^n e^{x_j} \quad (13A)$$

[0180] The SoftMax Layer Protocolは、入力 x を $x' = \log_2 e^x$ に置き換えて計算を行う。

よって、次のように式(13A)に対する証明を構成することができる。
 $x', y' \in \mathbb{R}$ に対して $y' = 2^{x'}$ が成立することを証明するプロトコルを構成する。そのプロトコルと(8) Multiplication Proofプロトコルと合わせることで、証明を構成することができる。

$y' = 2^{x'}$ が成立することを証明するプロトコルは、証明生成アルゴリズム $\text{Prove}_{\text{exp}}$ と、検証結果生成アルゴリズム $\text{Verify}_{\text{exp}}$ と、で構成される。

$\langle x' \rangle = x_0' \cdot 2^d + x_1'$ が成り立つ。本プロトコルは、式(13B)を計算する代わりに、8次多項式(13C)を用いて、式(13D)を計算する。

[0181] [数52]

$$2^{x_0' + x_1'/2^d} = 2^{x_0'} 2^{x_1'/2^d} \quad (13B)$$

$$P_{1045}(X) = c_8 X^8 + \dots + c_0 \quad (13C)$$

$$2^{x_1'} P_{1045}(x_0') \quad (13D)$$

[0182] ただし、 c_0 から c_8 は以下の通りである。

[0183]

[数53]

$$\begin{aligned}
c_0 &= 0.1000000000000077443021686 \cdot 10^1 \\
c_1 &= 0.693147180426163827795756 \cdot 10^0 \\
c_2 &= 0.240226510710170646053840 \cdot 10^0 \\
c_3 &= 0.555040686204663791577440 \cdot 10^{-1} \\
c_4 &= 0.961834122588046237497700 \cdot 10^{-2} \\
c_5 &= 0.133273035928143781932900 \cdot 10^{-2} \\
c_6 &= 0.155107460590052573978000 \cdot 10^{-3} \\
c_7 &= 0.141978473997656067110000 \cdot 10^{-4} \\
c_8 &= 0.186334772413796707600000 \cdot 10^{-5}
\end{aligned}$$

[0184] 証明生成アルゴリズム $\text{Prove}_{\text{exp}}$ を以下に記述する。

$x_0' [i]$ は、 x_0' の第 i ビットである。

[0185] [数54]

$\text{Prove}_{\text{exp}}(g, h, x')$:

$$z_1 := \langle x_1'^1 \rangle, \dots, z_8 := \langle x_1'^8 \rangle,$$

$$u := \langle 2^{x_0'} \rangle,$$

$$d := \langle P_{1045}(x_1') \rangle,$$

$$r_{\text{int}}, r_{\text{int}}^{(0)}, \dots, r_{\text{int}}^{(l-1)}, r_{\text{real}}, r_{\text{real}}^{(2)}, \dots, r_{\text{real}}^{(8)}, r_d, r_u \leftarrow Z_p,$$

[0186] $\text{Prove}_{\text{exp}}$ は、(7) Range Proofs プロトコルと (8) Multiplication Proofs プロトコルと (3) 一般化 Schnorr プロトコルを使用して、以下の式に対する証明 P を作成する。

[0187] [数55]

$$\begin{aligned}
C_{int} &= g^{r_{int}} h^{x'_0}, \\
C_{int}^{(i)} &= g^{r_{int}^{(i)}} h^{x'_{0}[i]} \quad i \in \{0, \dots, l-1\}, \\
C_{real} &= g^{r_{real}} h^{\langle x'_1 \rangle}, \\
C_{real}^{(i)} &= g^{r_{real}^{(i)}} h^{z_i} \quad i \in \{2, \dots, 8\}, \\
C_d &= g^{r_d} h^d, \\
C_u &= g^{r_u} h^u, \\
\langle x'_0 \rangle &\in \{2^d, \dots, 2^{d+l-1}\}, \langle x'_1 \rangle \in \{0, \dots, 2^d - 1\}, \langle x' \rangle = \langle x'_0 \rangle + \langle x'_1 \rangle, \\
x_1 &= \prod_{i=0}^{l-1} x'_1[i] \cdot 2^i, \\
u &= \prod_{i=0}^{l-1} (x'_1[i] \cdot 2^{2^i} + 1 - x'_1[i]), \\
z_i &= z_{i-1} \cdot \langle x'_1 \rangle, z_1 = \langle x'_0 \rangle \quad i \in \{2, \dots, 8\}, \\
d &= c_8 \cdot z_8 + \dots + c_0 = \langle P_{1045}(x'_0) \rangle, \\
\langle y \rangle &= \langle u \rangle \cdot \langle d \rangle.
\end{aligned}$$

[0188] 検証アルゴリズム $\text{Verify}_{\text{exp}}$ を以下に記述する。

[0189] [数56]

$\text{Verify}_{\text{exp}}(P):$

$$V := \text{Verify}_{\text{GenShnorr}}(P)$$

[0190] The SoftMax Layer プロトコルは、 (P_f) 証明生成アルゴリズム $\text{Prove}_{\text{SoftMax}}$ と、 (V_f) 検証アルゴリズム $\text{Verify}_{\text{SoftMax}}$ と、で構成される。

[0191] $\text{Prove}_{\text{SoftMax}}$ は、 $\text{Prove}_{\text{exp}}$ 、 $\text{Prove}_{\text{Mult}}$ を使用して、以下の式に対する証明 P を算出する。

[0192] [数57]

$\text{Prove}_{\text{SoftMax}}(g, h, x, y):$

$$y_i' = 2^{x_i'} \quad i \in \{1, \dots, n\},$$

$$y_i = y_i' / \sum_{j=1}^n y_j'.$$

[0193] $\text{Verify}_{\text{SoftMax}}$ は、検証結果 V を出力する。

[0194] [数58]

$\text{Verify}_{\text{SoftMax}}(P):$

$$V := \text{Verify}_{\text{GenShnorr}}(P)$$

[0195] ***実施の形態1の特徴***

実施の形態1は、以下のような特徴を有する。

実施の形態1は、推論モデルのモデルパラメータを整数値化し、推論モデルの検証を行う。

各層のゼロ知識証明プロトコルのアルゴリズムは、実施の形態1の特徴である。

推論モデルの検証手法は、重みパラメータ（モデルパラメータ）の整数値化とゼロ知識証明プロトコルとを組み合わせ実現される。

[0196] カッコ内に符号を記して、実施の形態1の特徴を示す。

推論装置（400）は、推論処理の対象となるデータ（ x ）である小数値を整数値で表現し、前記整数値を畳み込みニューラルネットワークのパラメータとして扱って推論モデル（ M ）を実行して推論結果（ c ）を得る。

証明装置（500）は、前記推論結果を入力にして証明生成アルゴリズムを実行して証明（ P ）を得る。

検証装置（６００）は、前記証明を入力にして検証アルゴリズムを実行して検証結果（Ｖ）を得る。

[0197] 前記推論結果が、前記畳み込みニューラルネットワークの各層の計算結果を含む。

証明装置（５００）は、前記畳み込みニューラルネットワークの層ごとに、前記層の前記計算結果を入力にして前記層の種類に応じたプロトコルの前記証明生成アルゴリズムを実行する。

前記証明が、前記畳み込みニューラルネットワークの各層に対する前記証明生成アルゴリズムの実行結果を含む。

検証装置（６００）は、前記畳み込みニューラルネットワークの層ごとに、前記層の前記実行結果を入力にして前記層の種類に応じた前記プロトコルの前記検証アルゴリズムを実行する。

前記検証結果が、前記畳み込みニューラルネットワークの各層に対する前記検証アルゴリズムの実行結果を含む。

[0198] ***実施の形態１の効果***

実施の形態１は、小数の固定小数点表示を整数値で表示することにより、小数のパラメータを扱うことが可能なゼロ知識証明のプロトコルを実現する。

これにより、パラメータが小数である推論モデルに対して、推論モデルの検証を行うことが可能になる。

[0199] 実施の形態１は、例えば以下のような効果を奏する。

データが第三者によって分析される。第三者は、機械学習モデルによる推論サービスを提供する者である。

実施の形態１により、分析されたデータの推論結果が実際に機械学習モデルを使ってデータの推論が行われることによって得られた結果であること、を推論モデルに関する情報を開示することなく証明することができる。

[0200] 実施の形態１の方法は、小数の固定小数点表示を整数値で表示し、離散対数問題の困難性を利用したゼロ知識証明を用いる。これにより、小数のパラ

メータを扱うことが可能なゼロ知識証明のプロトコルが実現される。そして、パラメータが小数であるような推論モデルに対して、推論モデルの検証を行うことが可能になる。

[0201] ***実施の形態１の補足***

パラメータ生成装置２００と鍵生成装置３００と推論装置４００と証明装置５００は、互いに組み合わされてもよい。つまり、推論検証システム１００は、パラメータ生成装置２００と鍵生成装置３００と推論装置４００と証明装置５００として機能する１台以上のコンピュータを備えてもよい。

[0202] パラメータ生成装置２００の生成部２２０は、公開パラメータ p を生成するために、乱数生成機能などを有してもよい。

鍵生成装置３００の生成部３２０は、公開鍵 p と秘密鍵 s を生成するために、乱数生成機能などを有してもよい。

[0203] 図１２に基づいて、パラメータ生成装置２００のハードウェア構成を説明する。

パラメータ生成装置２００は処理回路２０９を備える。

処理回路２０９は、受付部２１０と生成部２２０と出力部２３０を実現する処理回路である。

[0204] 処理回路は、専用のハードウェアであってもよいし、メモリに格納されるプログラムを実行するプロセッサであってもよい。

処理回路が専用のハードウェアである場合、処理回路は、例えば、単一回路、複合回路、プログラム化したプロセッサ、並列プログラム化したプロセッサ、ASIC、FPGAまたはこれらの組み合わせである。

ASICは、Application Specific Integrated Circuitの略称である。

FPGAは、Field Programmable Gate Arrayの略称である。

[0205] パラメータ生成装置２００は、処理回路２０９を代替する複数の処理回路を備えてもよい。

[0206] 処理回路 209 において、一部の機能が専用のハードウェアで実現されて、残りの機能がソフトウェアまたはファームウェアで実現されてもよい。

[0207] このように、パラメータ生成装置 200 の機能はハードウェア、ソフトウェア、ファームウェアまたはこれらの組み合わせで実現することができる。

[0208] 図 13 に基づいて、鍵生成装置 300 のハードウェア構成を説明する。

鍵生成装置 300 は処理回路 309 を備える。

処理回路 309 は、受付部 310 と生成部 320 と出力部 330 を実現する処理回路である。

[0209] 鍵生成装置 300 は、処理回路 309 を代替する複数の処理回路を備えてもよい。

[0210] 処理回路 309 において、一部の機能が専用のハードウェアで実現されて、残りの機能がソフトウェアまたはファームウェアで実現されてもよい。

[0211] このように、鍵生成装置 300 の機能はハードウェア、ソフトウェア、ファームウェアまたはこれらの組み合わせで実現することができる。

[0212] 図 14 に基づいて、推論装置 400 のハードウェア構成を説明する。

推論装置 400 は処理回路 409 を備える。

処理回路 409 は、受付部 410 と推論部 420 と出力部 430 を実現する処理回路である。

[0213] 推論装置 400 は、処理回路 409 を代替する複数の処理回路を備えてもよい。

[0214] 処理回路 409 において、一部の機能が専用のハードウェアで実現されて、残りの機能がソフトウェアまたはファームウェアで実現されてもよい。

[0215] このように、推論装置 400 の機能はハードウェア、ソフトウェア、ファームウェアまたはこれらの組み合わせで実現することができる。

[0216] 図 15 に基づいて、証明装置 500 のハードウェア構成を説明する。

証明装置 500 は処理回路 509 を備える。

処理回路 509 は、受付部 510 と保管部 520 と証明部 530 と出力部 540 を実現する処理回路である。

- [0217] 証明装置 500 は、処理回路 509 を代替する複数の処理回路を備えてもよい。
- [0218] 処理回路 509 において、一部の機能が専用のハードウェアで実現されて、残りの機能がソフトウェアまたはファームウェアで実現されてもよい。
- [0219] このように、証明装置 500 の機能はハードウェア、ソフトウェア、ファームウェアまたはこれらの組み合わせで実現することができる。
- [0220] 図 16 に基づいて、検証装置 600 のハードウェア構成を説明する。
検証装置 600 は処理回路 609 を備える。
処理回路 609 は、受付部 610 と保管部 620 と検証部 630 と出力部 640 を実現する処理回路である。
- [0221] 検証装置 600 は、処理回路 609 を代替する複数の処理回路を備えてもよい。
- [0222] 処理回路 609 において、一部の機能が専用のハードウェアで実現されて、残りの機能がソフトウェアまたはファームウェアで実現されてもよい。
- [0223] このように、検証装置 600 の機能はハードウェア、ソフトウェア、ファームウェアまたはこれらの組み合わせで実現することができる。
- [0224] 実施の形態 1 は、好ましい形態の例示であり、本開示の技術的範囲を制限することを意図するものではない。実施の形態 1 は、部分的に実施してもよいし、他の形態と組み合わせて実施してもよい。フローチャート等を用いて説明した手順は、適宜に変更してもよい。
- [0225] 推論検証システム 100 の各要素の「部」は、「処理」、「工程」、「回路」または「サーキットリ」と読み替えてもよい。

符号の説明

- [0226] 100 推論検証システム、200 パラメータ生成装置、201 プロセッサ、202 メモリ、203 補助記憶装置、204 通信装置、205 入出力インタフェース、209 処理回路、210 受付部、220 生成部、230 出力部、290 記憶部、300 鍵生成装置、301 プロセッサ、302 メモリ、303 補助記憶装置、304 通信装置、

305 入出力インタフェース、309 処理回路、310 受付部、320 生成部、330 出力部、390 記憶部、400 推論装置、401 プロセッサ、402 メモリ、403 補助記憶装置、404 通信装置、405 入出力インタフェース、409 処理回路、410 受付部、420 推論部、430 出力部、490 記憶部、500 証明装置、501 プロセッサ、502 メモリ、503 補助記憶装置、504 通信装置、505 入出力インタフェース、509 処理回路、510 受付部、520 保管部、521 鍵保管部、522 推論結果保管部、530 証明部、540 出力部、590 記憶部、600 検証装置、601 プロセッサ、602 メモリ、603 補助記憶装置、604 通信装置、605 入出力インタフェース、609 処理回路、610 受付部、620 保管部、621 鍵保管部、622 証明保管部、630 検証部、640 出力部、690 記憶部、c 推論結果、D パラメータ、M 推論モデル、P 証明、pk 公開鍵、pp 公開パラメータ、sk 秘密鍵、V 検証結果、x データ、 λ パラメータ。

請求の範囲

- [請求項1] 推論処理の対象となるデータである小数値を整数値で表現し、前記整数値を畳み込みニューラルネットワークのパラメータとして扱って推論モデルを実行して推論結果を得る推論部と、
- 前記推論結果を入力にして証明生成アルゴリズムを実行して証明を得る証明部と、
- 前記証明を入力にして検証アルゴリズムを実行して検証結果を得る検証部と、
- を備える推論検証システム。
- [請求項2] 前記推論結果が、前記畳み込みニューラルネットワークの各層の計算結果を含み、
- 前記証明部は、前記畳み込みニューラルネットワークの層ごとに、前記層の前記計算結果を入力にして前記層の種類に応じたプロトコルの前記証明生成アルゴリズムを実行し、
- 前記証明が、前記畳み込みニューラルネットワークの各層に対する前記証明生成アルゴリズムの実行結果を含み、
- 前記検証部は、前記畳み込みニューラルネットワークの層ごとに、前記層の前記実行結果を入力にして前記層の種類に応じた前記プロトコルの前記検証アルゴリズムを実行し、
- 前記検証結果が、前記畳み込みニューラルネットワークの各層に対する前記検証アルゴリズムの実行結果を含む
- 請求項1に記載の推論検証システム。
- [請求項3] The ReLU Layerプロトコルが、前記畳み込みニューラルネットワークのReLU活性化層に応じた前記プロトコルである
- 請求項2に記載の推論検証システム。
- [請求項4] The Affine Layerプロトコルが、前記畳み込みニューラルネットワークのAffine層に応じた前記プロトコルである

請求項2または請求項3に記載の推論検証システム。

[請求項5] The Convolution Layerプロトコルが、前記畳み込みニューラルネットワークの畳み込み層に応じた前記プロトコルである

請求項2から請求項4のいずれか1項に記載の推論検証システム。

[請求項6] The Average Pooling Layerプロトコルが、前記畳み込みニューラルネットワークのAverage Pooling層に応じた前記プロトコルである

請求項2から請求項5のいずれか1項に記載の推論検証システム。

[請求項7] The Max Pooling Layerプロトコルが、前記畳み込みニューラルネットワークのMax Pooling層に応じた前記プロトコルである

請求項2から請求項6のいずれか1項に記載の推論検証システム。

[請求項8] The SoftMax Layerプロトコルが、前記畳み込みニューラルネットワークのSoftMax層に応じた前記プロトコルである

請求項2から請求項7のいずれか1項に記載の推論検証システム。

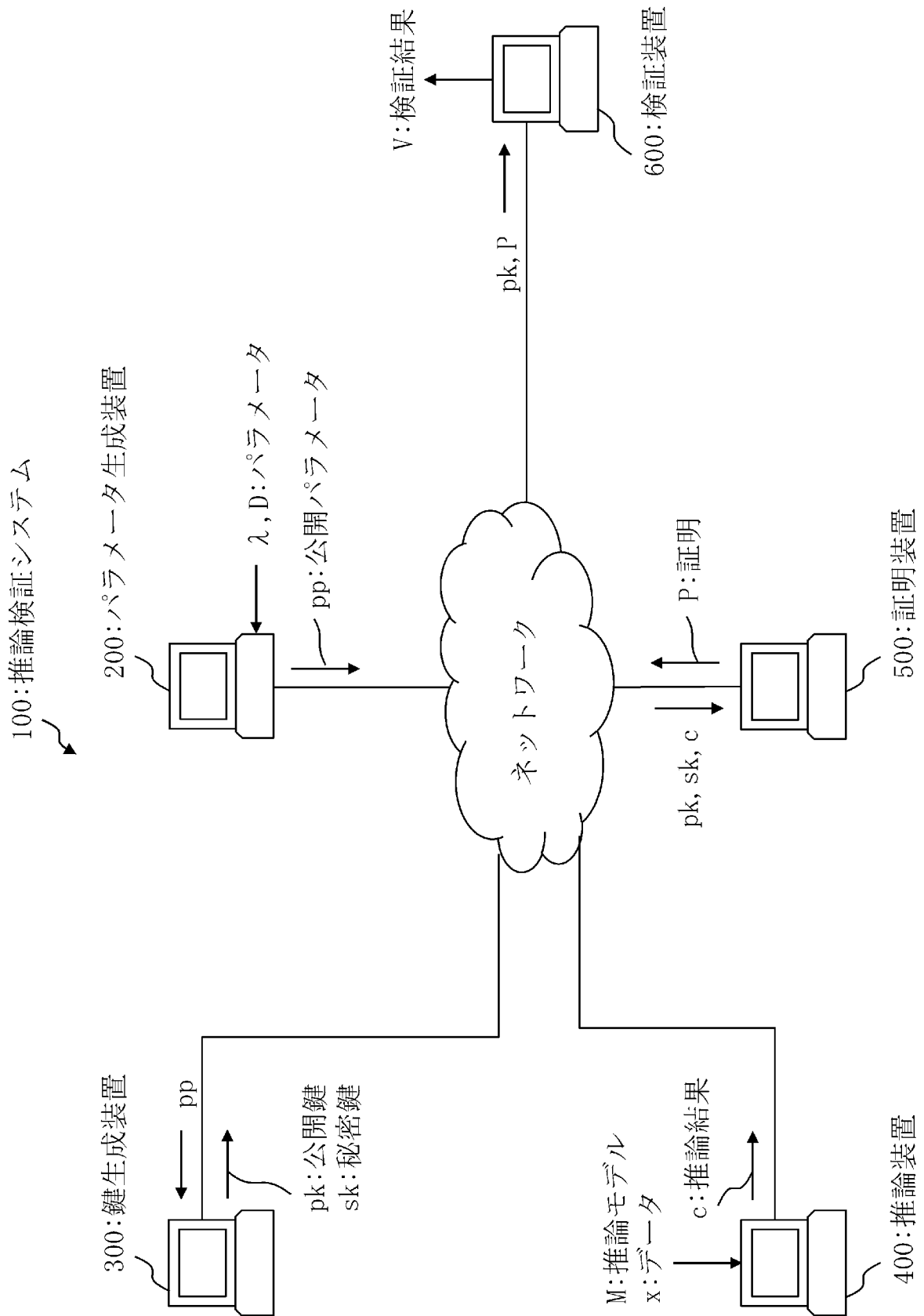
[請求項9] 推論処理の対象となるデータである小数値を整数値で表現し、前記整数値を畳み込みニューラルネットワークのパラメータとして扱って推論モデルを実行して推論結果を得て、

前記推論結果を入力にして証明生成アルゴリズムを実行して証明を得て、

前記証明を入力にして検証アルゴリズムを実行して検証結果を得る推論検証方法。

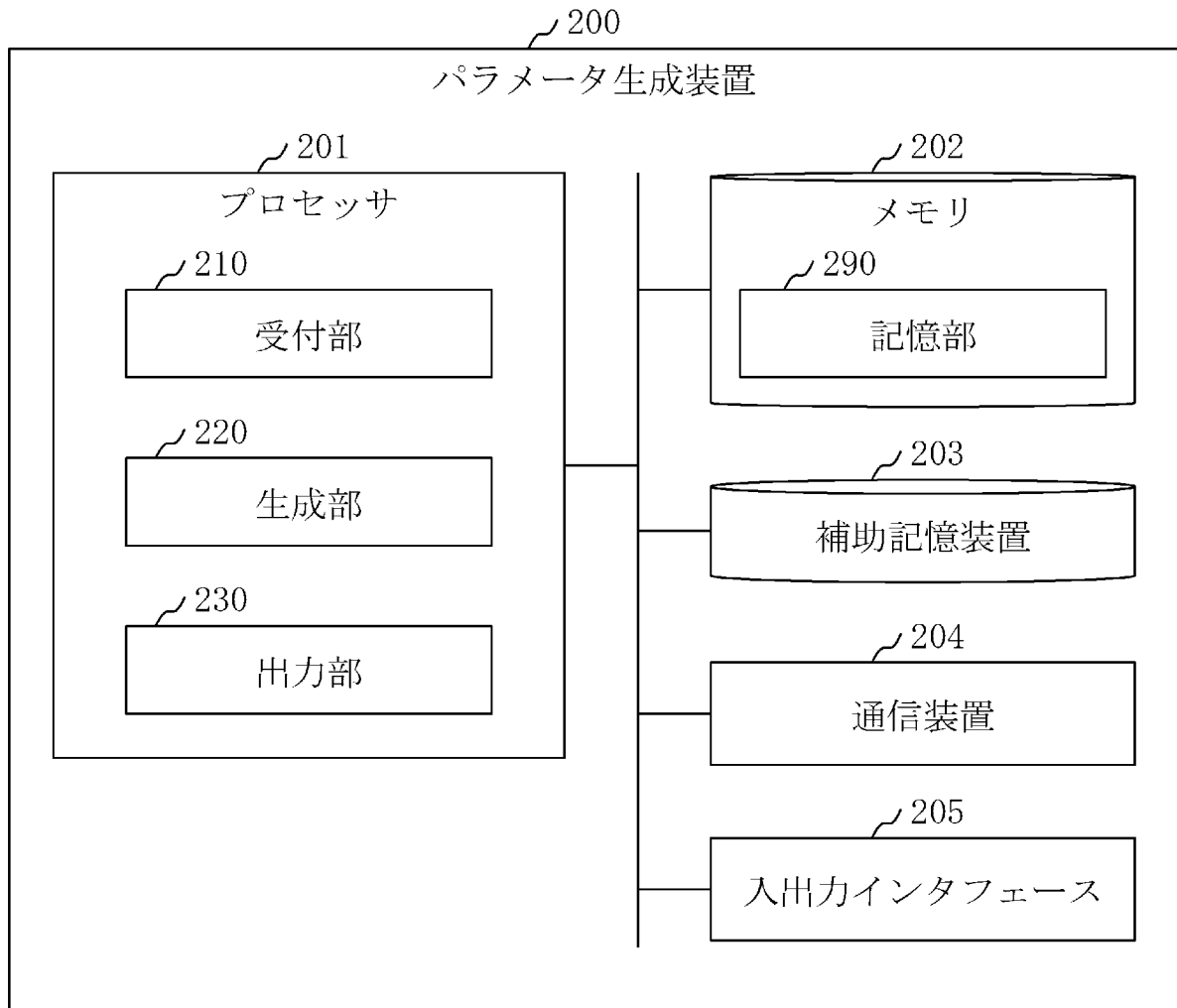
[図1]

図1



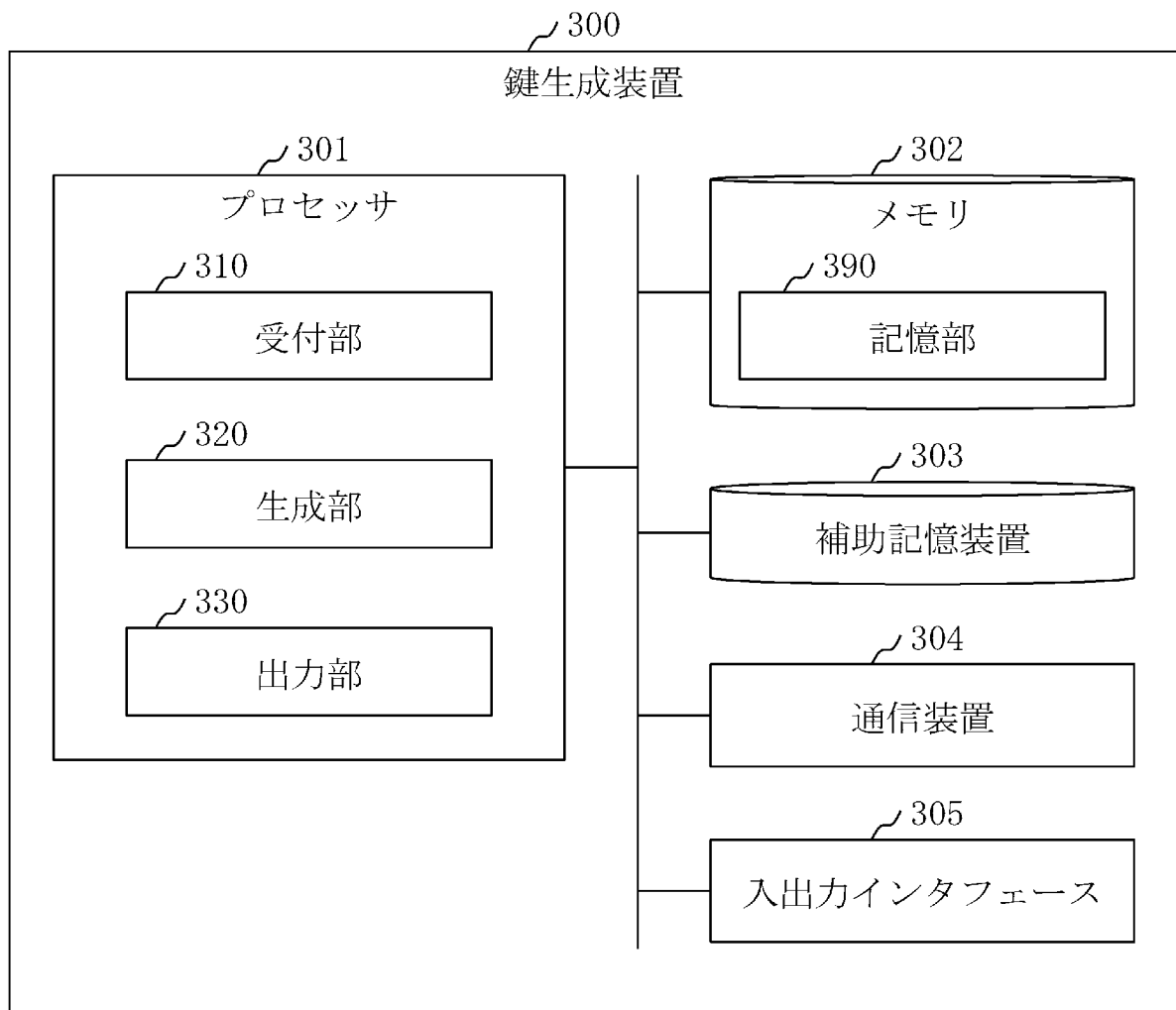
[図2]

図2



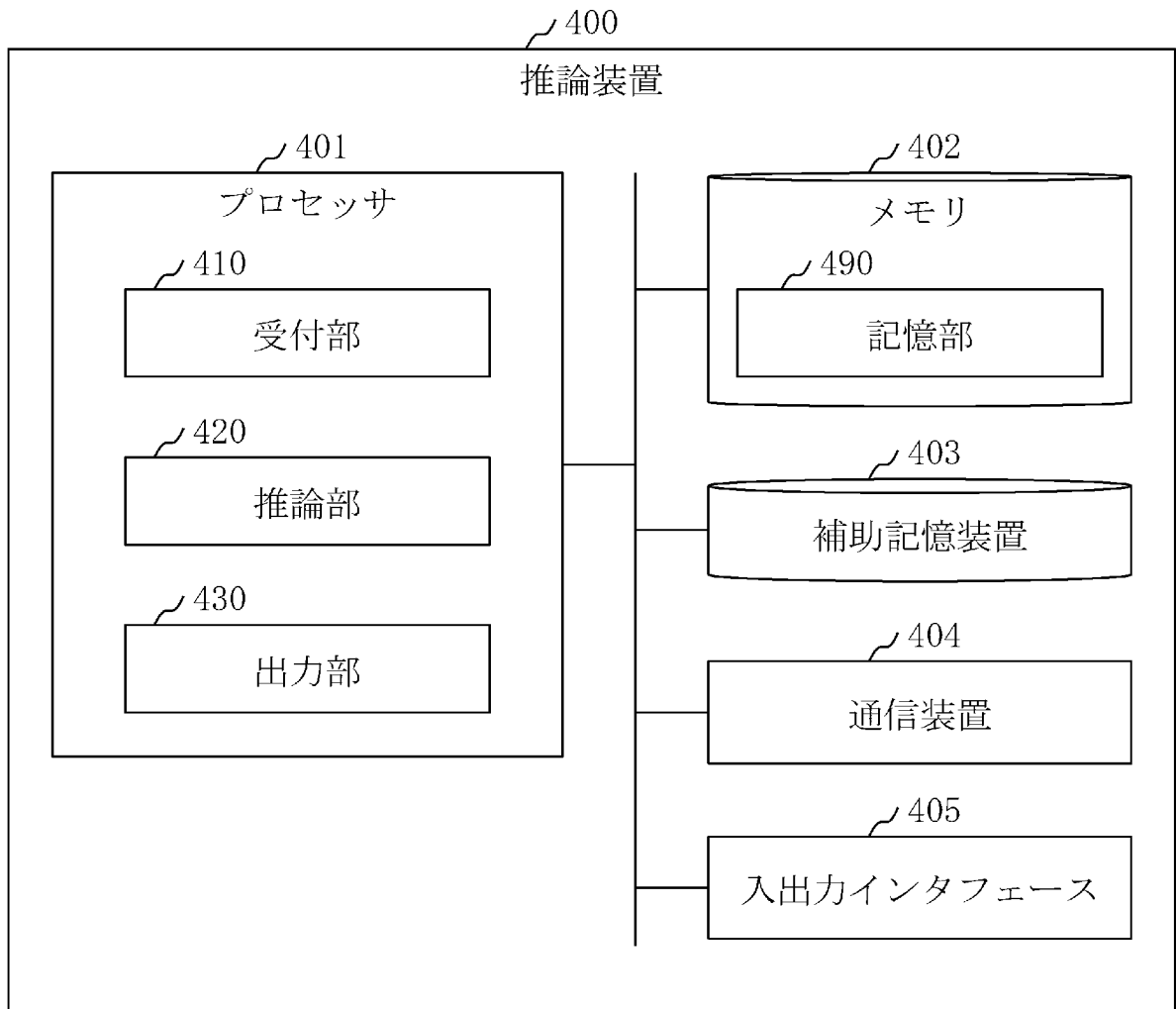
[図3]

図3



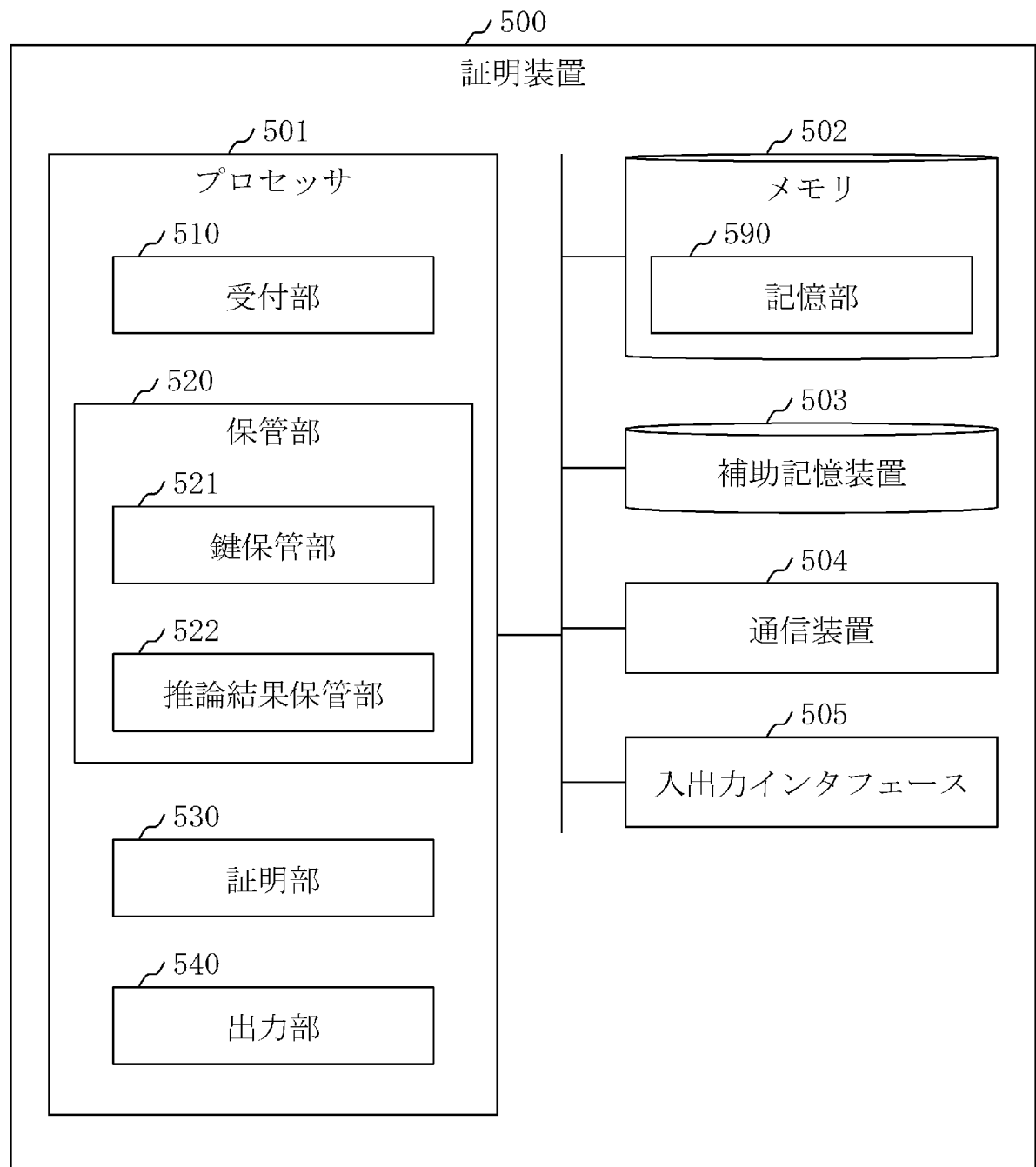
[図4]

図4



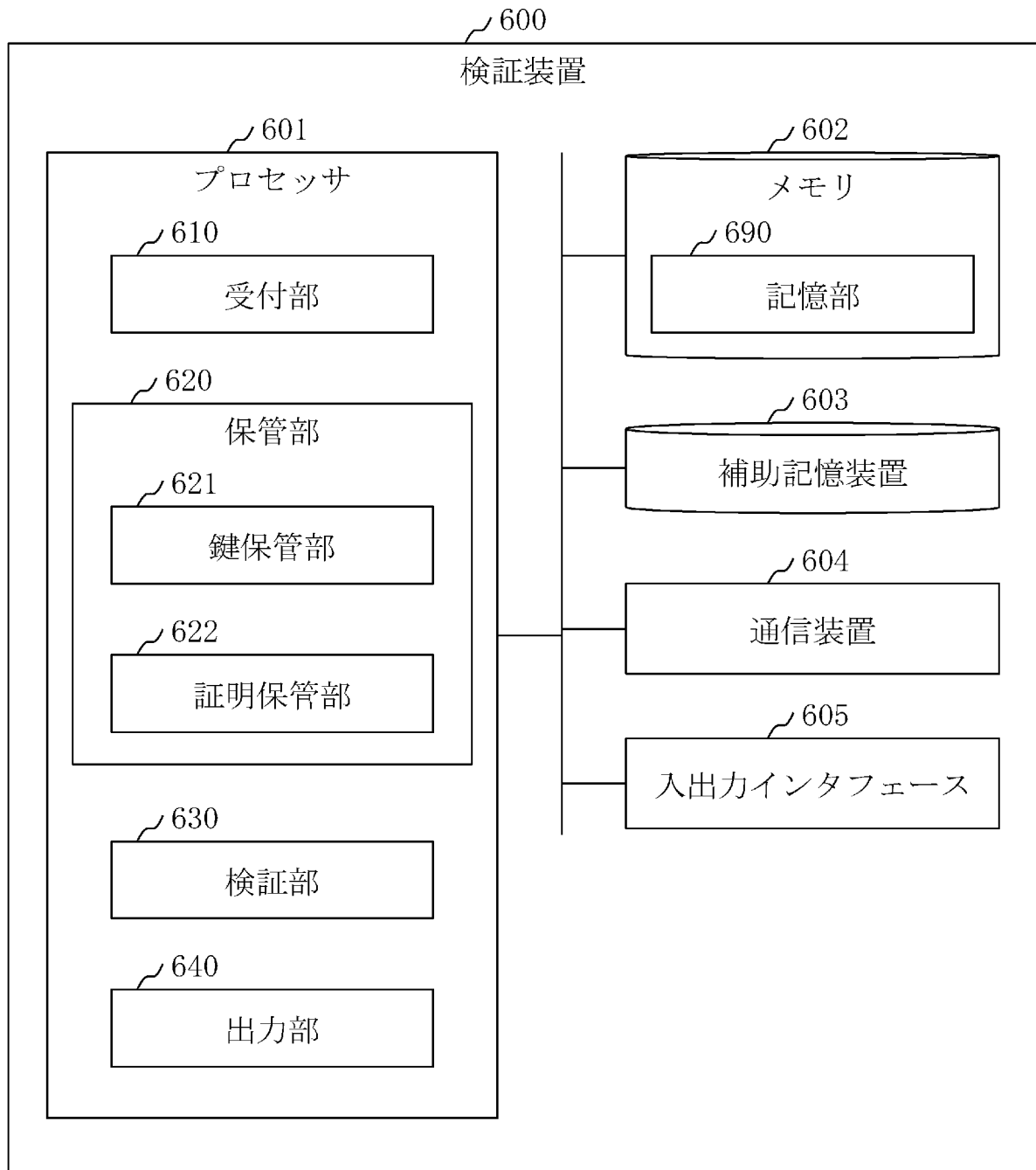
[図5]

図5



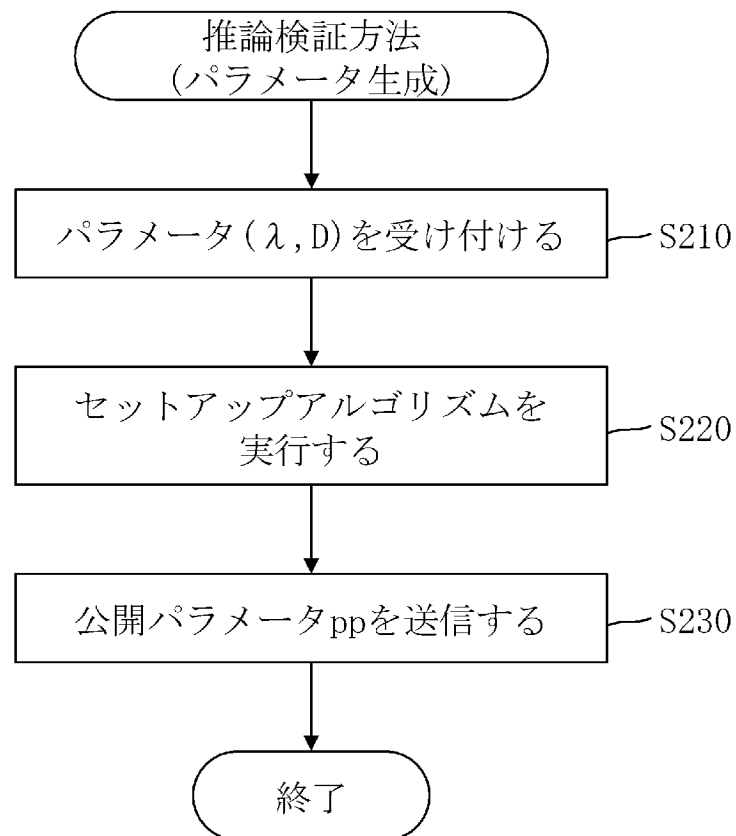
[図6]

図6



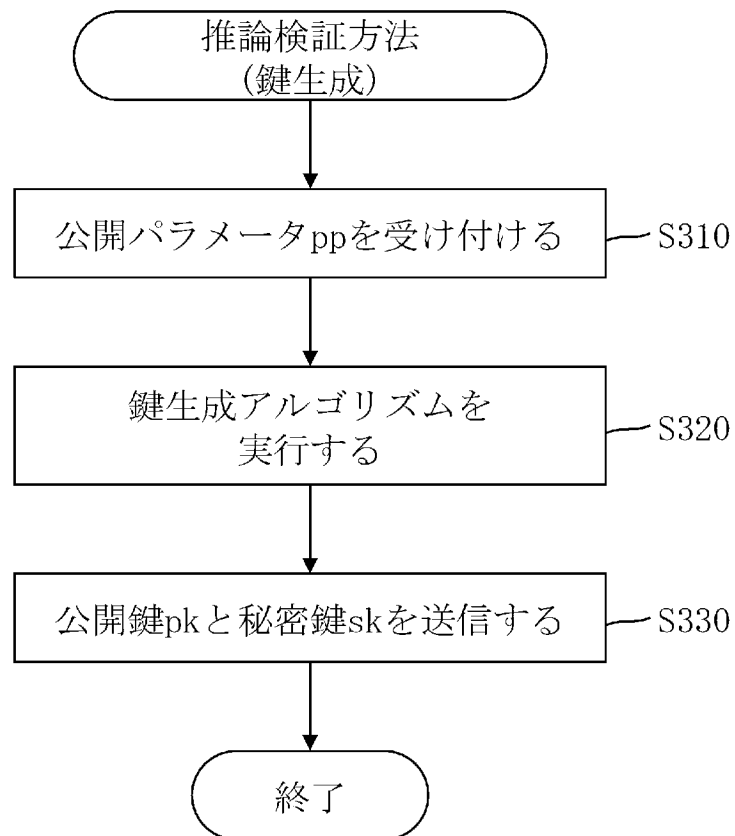
[図7]

図7



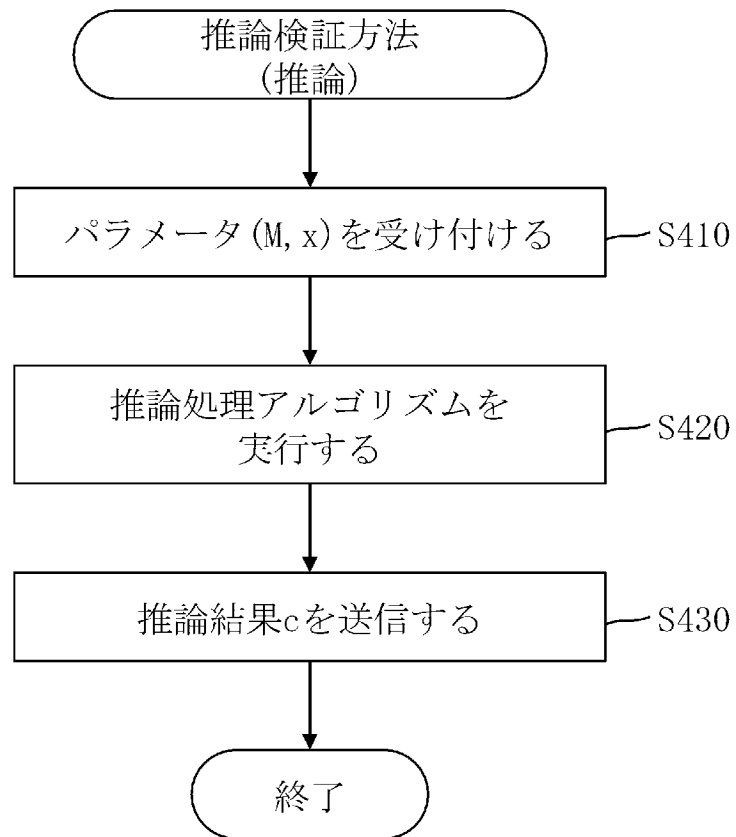
[図8]

図8



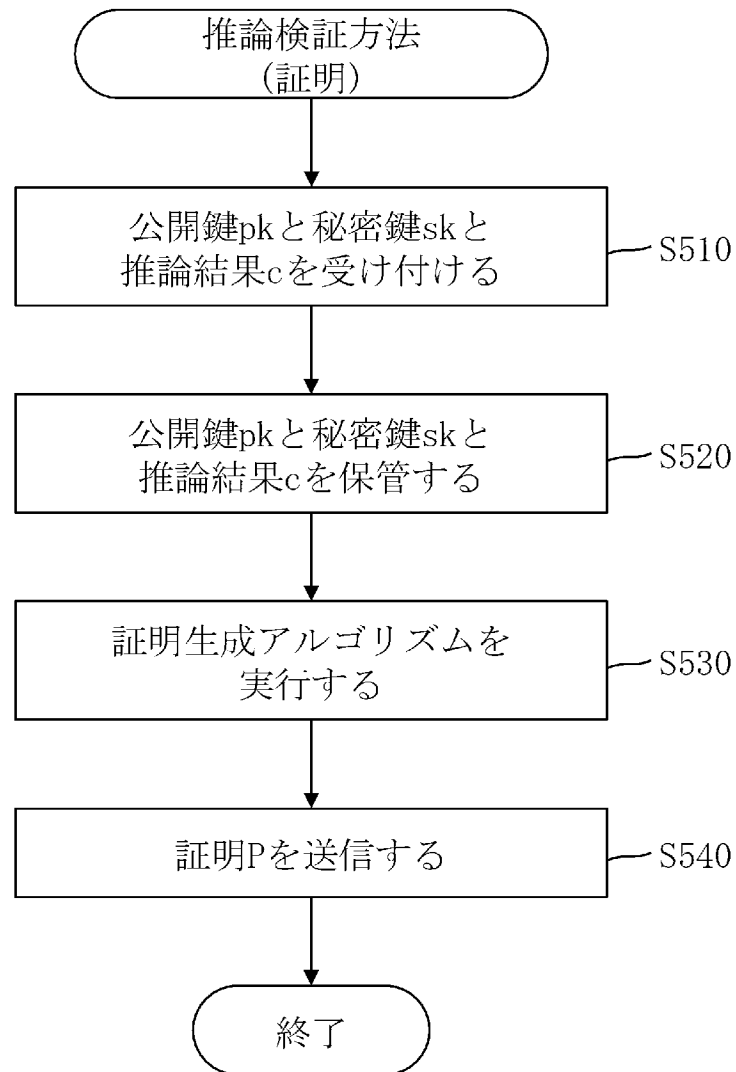
[図9]

図9



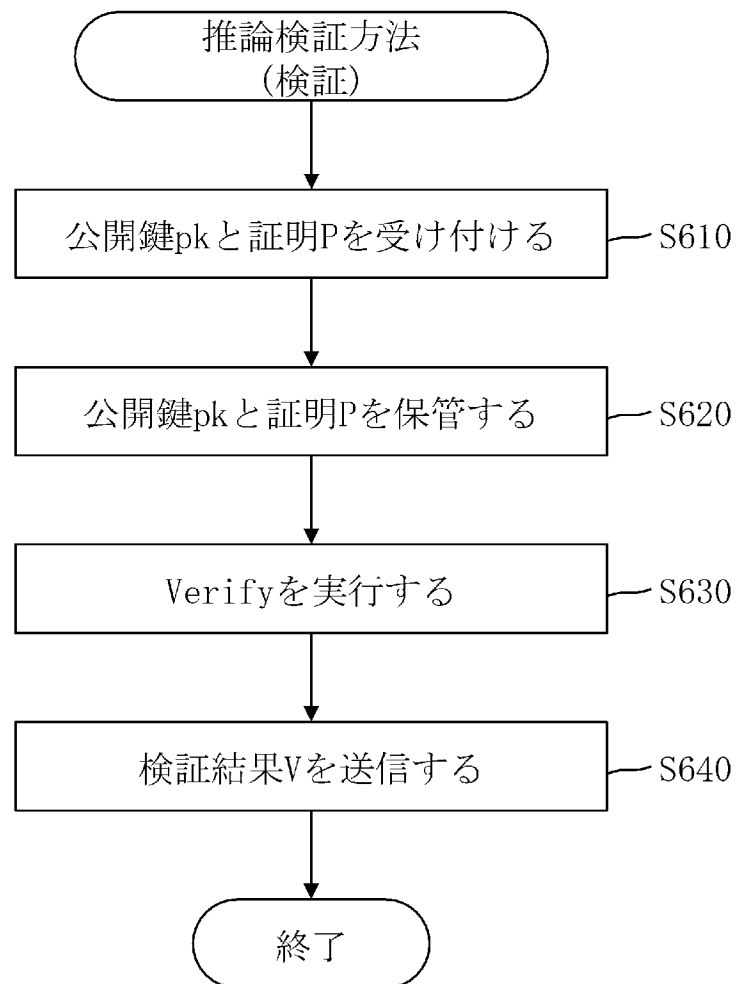
[図10]

図10



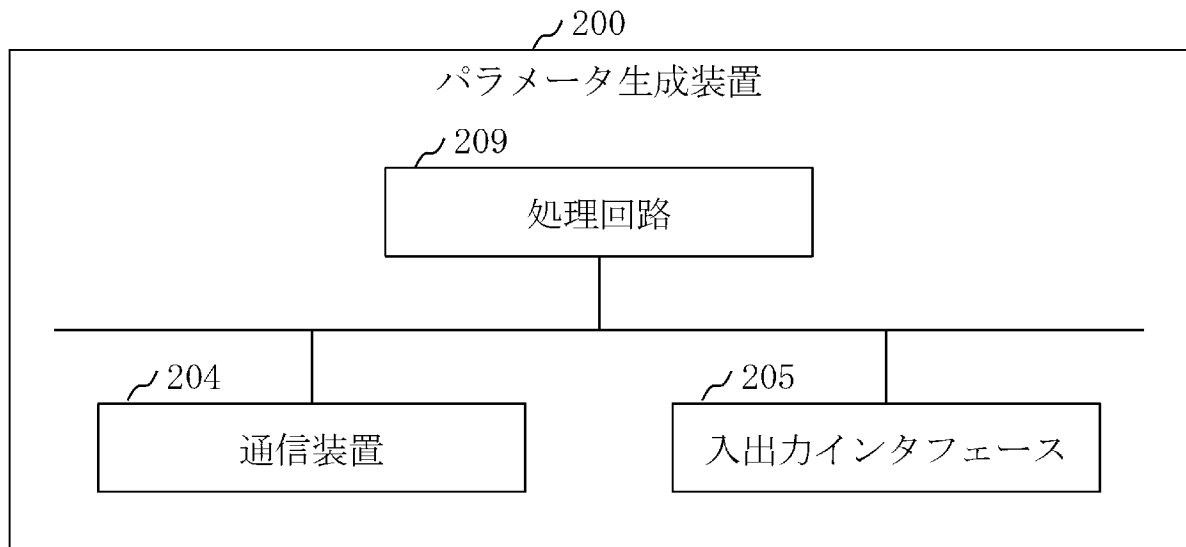
[図11]

図11



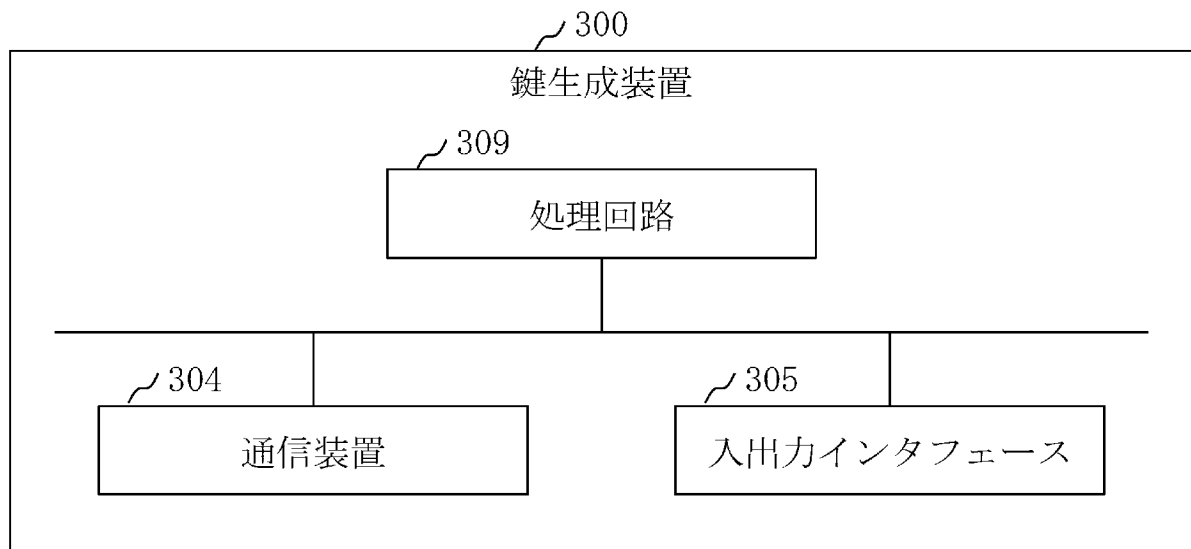
[図12]

図12



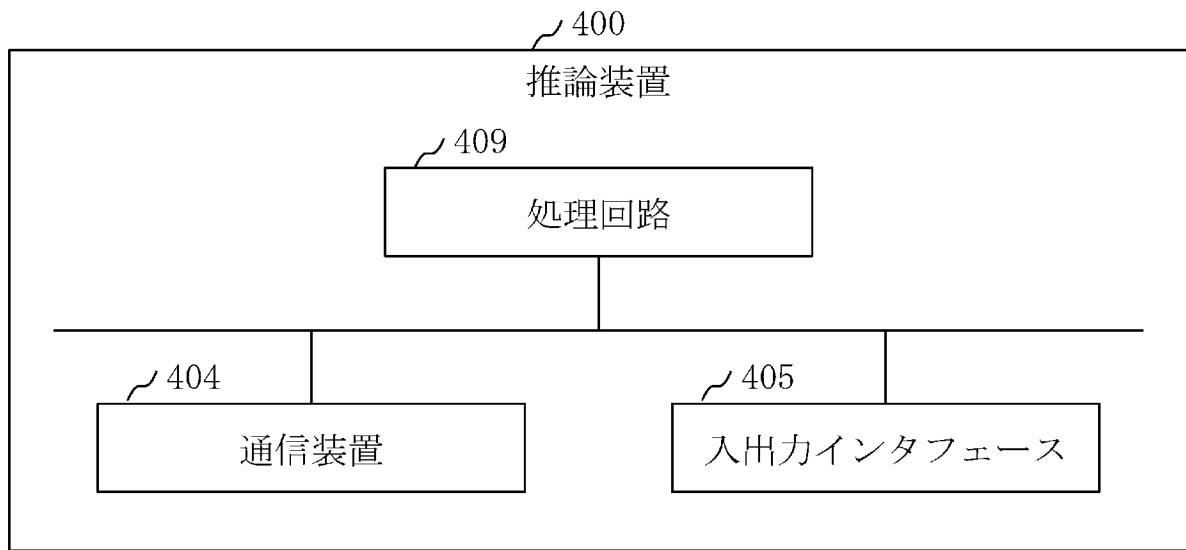
[図13]

図13



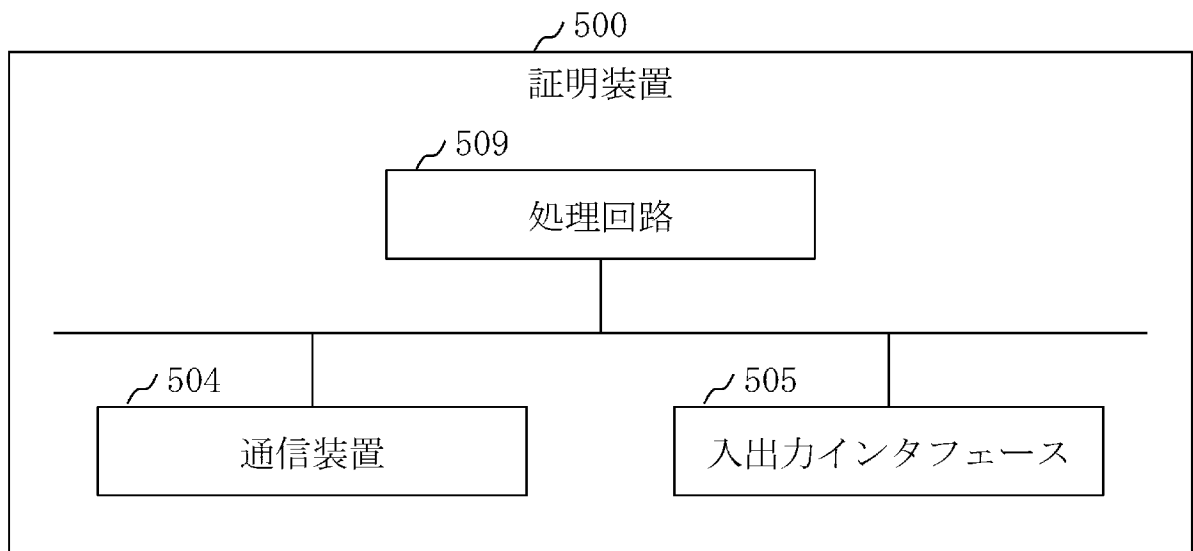
[図14]

図14



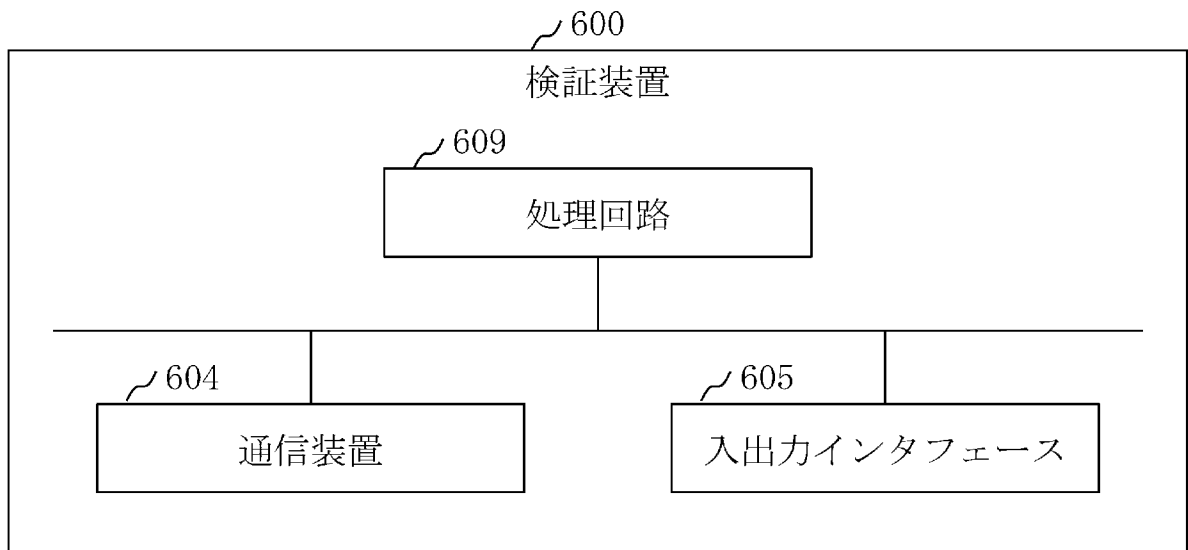
[図15]

図15



[図16]

図16



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2022/026498

A. CLASSIFICATION OF SUBJECT MATTER*G06N 3/02*(2006.01)i

FI: G06N3/02

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06N3/02

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Published examined utility model applications of Japan 1922-1996

Published unexamined utility model applications of Japan 1971-2022

Registered utility model specifications of Japan 1996-2022

Published registered utility model applications of Japan 1994-2022

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	FENG, Boyuan et al. ZEN: An Optimizing Compiler for Verifiable, Zero-Knowledge Neural Network Inferences. Cryptology ePrint Archive. 15 May 2021, [online], [retrieved on 16 September 2022], Retrieved from the Internet: <URL: https://eprint.iacr.org/archive/2021/087/20210515:224258 >	1, 9
A	abstract, sections 1, 3, 4	2-8
A	US 2021/0406436 A1 (INDUSTRY-UNIVERSITY COOPERATION FOUNDATION HANYANG UNIVERSITY) 30 December 2021 (2021-12-30)	1-9



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search

16 September 2022

Date of mailing of the international search report

27 September 2022

Name and mailing address of the ISA/JP

Japan Patent Office (ISA/JP)

3-4-3 Kasumigaseki, Chiyoda-ku, Tokyo 100-8915

Japan

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/JP2022/026498

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
US	2021/0406436	A1	30 December 2021	KR 10-2022-0002046	A

A. 発明の属する分野の分類（国際特許分類（IPC）） G06N 3/02(2006.01)i FI: G06N3/02										
B. 調査を行った分野										
調査を行った最小限資料（国際特許分類（IPC）） G06N3/02										
最小限資料以外の資料で調査を行った分野に含まれるもの <table border="0"> <tr> <td>日本国実用新案公報</td> <td>1922 - 1996年</td> </tr> <tr> <td>日本国公開実用新案公報</td> <td>1971 - 2022年</td> </tr> <tr> <td>日本国実用新案登録公報</td> <td>1996 - 2022年</td> </tr> <tr> <td>日本国登録実用新案公報</td> <td>1994 - 2022年</td> </tr> </table>			日本国実用新案公報	1922 - 1996年	日本国公開実用新案公報	1971 - 2022年	日本国実用新案登録公報	1996 - 2022年	日本国登録実用新案公報	1994 - 2022年
日本国実用新案公報	1922 - 1996年									
日本国公開実用新案公報	1971 - 2022年									
日本国実用新案登録公報	1996 - 2022年									
日本国登録実用新案公報	1994 - 2022年									
国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）										
C. 関連すると認められる文献										
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号								
X	FENG, Boyuan et al., ZEN: An Optimizing Compiler for Verifiable, Zero-Knowledge Neural Network Inferences, Cryptology ePrint Archive, 2021.05.15, [online], [retrieved on 2022.09.16], Retrieved from the Internet: <URL: https://eprint.iacr.org/archive/2021/087/20210515:224258 > abstract, Sec. 1, 3, 4	1, 9								
A		2-8								
A	US 2021/0406436 A1 (INDUSTRY-UNIVERSITY COOPERATION FOUNDATION HANYANG UNIVERSITY) 30.12.2021 (2021 - 12 - 30)	1-9								
☐ C欄の続きにも文献が列挙されている。 ☒ パテントファミリーに関する別紙を参照。										
<table border="0"> <tr> <td> * 引用文献のカテゴリー “A” 特に関連のある文献ではなく、一般的技术水準を示すもの “E” 国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの “L” 優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す） “O” 口頭による開示、使用、展示等に言及する文献 “P” 国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献 </td> <td> “T” 国際出願日又は優先日後に公表された文献であって出願と抵触するものではなく、発明の原理又は理論の理解のために引用するもの “X” 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの “Y” 特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの “&” 同一パテントファミリー文献 </td> </tr> </table>			* 引用文献のカテゴリー “A” 特に関連のある文献ではなく、一般的技术水準を示すもの “E” 国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの “L” 優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す） “O” 口頭による開示、使用、展示等に言及する文献 “P” 国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献	“T” 国際出願日又は優先日後に公表された文献であって出願と抵触するものではなく、発明の原理又は理論の理解のために引用するもの “X” 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの “Y” 特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの “&” 同一パテントファミリー文献						
* 引用文献のカテゴリー “A” 特に関連のある文献ではなく、一般的技术水準を示すもの “E” 国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの “L” 優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す） “O” 口頭による開示、使用、展示等に言及する文献 “P” 国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献	“T” 国際出願日又は優先日後に公表された文献であって出願と抵触するものではなく、発明の原理又は理論の理解のために引用するもの “X” 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの “Y” 特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの “&” 同一パテントファミリー文献									
国際調査を完了した日 16.09.2022	国際調査報告の発送日 27.09.2022									
名称及びあて先 日本国特許庁(ISA/JP) 〒100-8915 日本国 東京都千代田区霞が関三丁目4番3号	権限のある職員（特許庁審査官） 大桃 由紀雄 5B 5583 電話番号 03-3581-1101 内線 3545									

国際調査報告
パテントファミリーに関する情報

国際出願番号

PCT/JP2022/026498

引用文献			公表日	パテントファミリー文献			公表日
US	2021/0406436	A1	30.12.2021	KR	10-2022-0002046	A	