(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7621675号
(P7621675)

(45)発行日　令和7年1月27日(2025.1.27)

(24)登録日　令和7年1月17日(2025.1.17)

(51)国際特許分類

		FI		
G06F 21/60 (2013.01)		G06F	21/60	320
G06F 21/62 (2013.01)		G06F	21/62	318
G06N 3/02 (2006.01)		G06N	3/02	
G06N 3/0464(2023.01)		G06N	3/0464	

請求項の数　6　（全25頁）

(21)出願番号　特願2023-129058(P2023-129058)
(22)出願日　令和5年8月8日(2023.8.8)
(65)公開番号　特開2024-160912(P2024-160912A)
(43)公開日　令和6年11月15日(2024.11.15)
審査請求日　令和5年8月8日(2023.8.8)
(31)優先権主張番号　10-2023-0058605
(32)優先日　令和5年5月4日(2023.5.4)
(33)優先権主張国・地域又は機関
韓国(KR)
特許法第３０条第２項適用 ウェブサイトの掲載日 ２０２２年１０月２４日 ウェブサイトのアドレス ｈｔｔｐｓ：／／ｄｌ．ａｃｍ．ｏｒｇ／ｄｏｉ／ａｂｓ／１０．１１４５／３５２８５３５．３５３１５１３

(73)特許権者　514274672
延世大学校 産学協力団
ＵＩＦ（Ｕｎｉｖｅｒｓｉｔｙ Ｉｎｄｕｓｔｒｙ Ｆｏｕｎｄａｔｉｏｎ），Ｙｏｎｓｅｉ Ｕｎｉｖｅｒｓｉｔｙ
大韓民国 ０３７２２ ソウル、 ソデムーグ、 ヨンセイ－ロ ５０５０，ＹＯＮＳＥＩ－ＲＯ，ＳＥＯＤＡＥＭＵＮ－ＧＵ，ＳＥＯＵＬ ０３７２２，ＲＥＰＵＢＬＩＣ ＯＦ ＫＯＲＥＡ
(74)代理人　110000051
弁理士法人共生国際特許事務所
(72)発明者　キム，ヨン ソク
大韓民国 ソウル特別市 西大門区 延世路
最終頁に続く

(54)【発明の名称】　トラスト環境ベースの人工知能装置

(57)【特許請求の範囲】

【請求項１】

暗号化入力データを送信し、暗号化出力データを受信する第１類型のメモリと、

トラスト信頼空間で動作し、前記暗号化入力データ及び前記暗号化出力データの人工知能演算を行うトラスト人工知能処理部と、を備え、

前記トラスト人工知能処理部は、

前記暗号化入力データの復号化を介して復号化入力データを生成し、前記暗号化出力データを生成するために非暗号化出力データの暗号化を行う暗号処理フロント－エンドプロセッサと、

前記復号化入力データ及び前記非暗号化出力データに対するバッファを提供し、前記第１類型のメモリよりも相対的に速い動作速度及び少ない格納容量を有する第２類型のメモリと、

前記復号化入力データに基づいてニューラルネットワーク演算を行って前記非暗号化出力データを生成するプロセッサと、を含み、

前記プロセッサは、

直接畳み込みベースのニューラルネットワーク演算を行って前記第１類型のメモリのアクセス数を減らし、

前記第２類型のメモリを、畳み込みレイヤの入力アクティベーション、復号化フィルタ、及び非暗号出力アクティベーションのアクセス領域により管理し、

前記第２類型のメモリに畳み込みレイヤの実行全体で繰り返しアクセスするときに、全

ての前記復号化入力データを前記入力アクティベーションの領域に固定的に格納し、残りの前記第２類型のメモリを、前記復号化フィルタ及び前記非暗号出力アクティベーションの領域の２つの循環キューで構成された循環キュー方式で格納し、
前記暗号処理フロント－エンドプロセッサの暗号化及び復号化演算の実行にオーバーラップするように前記ニューラルネットワーク演算を実行してイントラ－レイヤパイプライニングを実現し、
前記イントラ－レイヤパイプライニングのために、データ復号化ステップ、演算ステップ、及びデータ暗号化ステップに細分化して、前記ニューラルネットワーク演算を途切れることなく行うことを特徴とするトラスト環境ベースの人工知能装置。
【請求項２】
前記暗号処理フロント－エンドプロセッサは、前記第１類型のメモリから前記暗号化入力データとして暗号化入力アクティベーション及び暗号化フィルタの入力を受けて、前記第２類型のメモリに復号化入力アクティベーション及び復号化フィルタを格納することを特徴とする請求項１に記載のトラスト環境ベースの人工知能装置。
【請求項３】
前記暗号処理フロント－エンドプロセッサは、前記人工知能演算の過程で前記プロセッサによるオン－デマンド要求を受信し、前記第１類型のメモリにアクセスして前記暗号化入力データを取り込むことを特徴とする請求項１に記載のトラスト環境ベースの人工知能装置。
【請求項４】
前記プロセッサは、前記暗号処理フロント－エンドプロセッサの割り込み駆動のオフロード（ｉｎｔｅｒｒｕｐｔ　ｄｒｉｖｅｎ　ｏｆｆｌｏａｄｉｎｇ）を介して、前記第１類型のメモリ及び前記第２類型のメモリとデータ送受信を行うことを特徴とする請求項１に記載のトラスト環境ベースの人工知能装置。
【請求項５】
前記プロセッサは、ＤＭＡ（Ｄｉｒｅｃｔ　Ｍｅｍｏｒｙ　Ａｃｃｅｓｓ）コントローラのＤＭＡ駆動のオフロードを介して、前記暗号処理フロント－エンドプロセッサと前記第１類型のメモリ及び前記第２類型のメモリとデータ送受信を行うことを特徴とする請求項１に記載のトラスト環境ベースの人工知能装置。
【請求項６】
前記プロセッサは、前記ニューラルネットワーク演算の実行中に前記暗号処理フロント－エンドプロセッサが前記暗号化入力データの復号化演算を行うようにして、前記ニューラルネットワーク演算を途切れることなく行うことを特徴とする請求項１に記載のトラスト環境ベースの人工知能装置。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、人工知能ニューラルネットワークフレームワーク技術に関し、より詳細には、トラスト（Ｔｒｕｓｔ）環境で人工ニューラルネットワークの実行を安全に加速化することができるトラスト環境ベースの人工知能装置に関する。
【背景技術】
【０００２】
ＤＮＮ（Ｄｅｅｐ　Ｎｅｕｒａｌ　Ｎｅｔｗｏｒｋ）は、入力層及び出力層とその間のいくつかの隠れ層とからなる人工ニューラルネットワークである。ＤＮＮは、モバイル及び組み込みアプリケーションで広く使用されている。特に、ＤＮＮは、ユーザの身元を検証するためにユーザの生物学的特性（例えば、指紋、虹彩、顔など）を使用する生体認証実行のアプリケーションに有用である。
【０００３】
ＤＮＮには敏感なユーザデータが多く含まれているため、モバイル及び組み込み機器はセキュリティ攻撃からユーザ及びＤＮＮデータを安全に保護することができる安全なＤＮ

Ｎの実行環境を実現しなければならない。
【０００４】
従来は、ＡＲＭ（Ａｄｖａｎｃｅｄ ＲＩＳＣ Ｍａｃｈｉｎｅ）プロセッサで使用されるハードウェアベースのセキュリティ技術であるトラストゾーン（ＴｒｕｓｔＺｏｎｅ）を介してトラスト実行環境（Ｔｒｕｓｔｅｄ Ｅｘｅｃｕｔｉｏｎ Ｅｎｖｉｒｏｎｍｅｎｔ）でＤＮＮを実行することが提案されていた。トラストゾーンは、プロセッサ（ｐｒｏｃｅｓｓｏｒ）内に独立したセキュリティ区域を別々に配置して重要な情報を保護するハードウェアベースのセキュリティ技術である。しかし、トラストゾーンでＤＮＮを実行するだけではデータを完全に保護することはできない。トラストゾーンはメモリ保護機能が制限されているためである。トラストゾーンを使用すると、ハードウェア及びソフトウェアリソースを安全な一般区域に分割して、ＤＮＮの実行を他のプロセスから分離することができる。一般区域でセキュリティ区域のデータにアクセスすることは、ハードウェアによって防止される。しかし、トラストゾーンはデータを揮発性メモリに暗号化していない状態で維持するため、コールドブート攻撃（Ｃｏｌｄ Ｂｏｏｔ Ａｔｔａｃｋ）のような物理的なセキュリティ攻撃は、トラストゾーンでＤＮＮの実行にも拘らず、敏感なユーザ及びＤＮＮデータを取得することができる。
【０００５】
物理的攻撃からデータを保護するために、データを暗号化して安全なオンチップメモリでのみを復号化するように選択することができる。このような方式は、トラストゾーンを使用してＤＮＮの実行を他のプロセスから隔離することができるだけでなく、暗号化を介して物理的攻撃からユーザ及びＤＮＮデータを保護することができる。しかし、メモリで暗号化されたデータを保護すると、遅いメモリアクセスが大きく増加し、プロセッサに課される高いデータ暗復号化のオーバーヘッドにより、ＤＮＮの実行時間が大きく増加するという問題が発生する。
【０００６】
従って、敏感なユーザ及びＤＮＮデータを物理的攻撃から保護するだけではなく、遅いメモリアクセス及び高いデータ暗復号化のオーバーヘッドを克服し、ＤＮＮの実行時間を減らす新しいセキュリティＤＮＮフレームワークが必要になった。
【先行技術文献】
【特許文献】
【０００７】
【文献】韓国登録特許第１０－２４７４８７５号公報
【発明の概要】
【発明が解決しようとする課題】
【０００８】
本発明は、上記従来の問題点に鑑みてなされたものであって、本発明の目的は、トラスト（Ｔｒｕｓｔ）環境で人工ニューラルネットワークの実行を安全に加速化することができるトラスト環境ベースの人工知能装置を提供することにある。
【課題を解決するための手段】
【０００９】
上記目的を達成するためになされた本発明の一態様によるトラスト環境ベースの人工知能装置は、暗号化入力データを送信し、暗号化出力データを受信する第１類型のメモリと、トラスト信頼空間で動作し、前記暗号化入力データ及び前記暗号化出力データの人工知能演算を行うトラスト人工知能処理部と、を備え、前記トラスト人工知能処理部は、前記暗号化入力データの復号化を介して復号化入力データを生成し、前記暗号化出力データを生成するために非暗号化出力データの暗号化を行う暗号処理フロント－エンドプロセッサと、前記復号化入力データ及び前記非暗号化出力データに対するバッファを提供する第２類型のメモリと、前記復号化入力データに基づいてニューラルネットワーク演算を行って前記非暗号化出力データを生成するプロセッサと、を含む。
【００１０】

前記暗号処理フロント－エンドプロセッサは、前記第１類型のメモリから前記暗号化入力データとして暗号化入力アクティベーション及び暗号化フィルタの入力を受けて、前記第２類型のメモリに復号化入力アクティベーション及び復号化フィルタを格納することができる。

前記暗号処理フロント－エンドプロセッサは、前記人工知能演算の過程で前記プロセッサによるオン－デマンド要求を受信し、前記第１類型のメモリにアクセスして前記暗号化入力データを取り込むことができる。

前記第２類型のメモリは、前記第１類型のメモリよりも相対的に速い動作速度及び少ない格納容量を有することができる。

前記プロセッサは、直接畳み込みベースのニューラルネットワーク演算を行って前記第１類型のメモリのアクセス数を減らすことができる。

前記プロセッサは、前記第２類型のメモリに復号化入力アクティベーションを固定的に格納し、復号化フィルタ及び非暗号出力アクティベーションを循環キュー方式で格納することができる。

前記プロセッサは、前記暗号処理フロント－エンドプロセッサの割り込み駆動のオフロード（ｉｎｔｅｒｒｕｐｔ　ｄｒｉｖｅｎ　ｏｆｆｌｏａｄｉｎｇ）を介して、前記第１類型のメモリ及び前記第２類型のメモリとデータ送受信を行うことができる。

前記プロセッサは、ＤＭＡ（Ｄｉｒｅｃｔ　Ｍｅｍｏｒｙ　Ａｃｃｅｓｓ）コントローラのＤＭＡ駆動のオフロードを介して、前記暗号処理フロント－エンドプロセッサと前記第１類型のメモリ及び前記第２類型のメモリとデータ送受信を行うことができる。

前記プロセッサは、前記暗号処理フロント－エンドプロセッサの暗号化及び復号化演算の実行にオーバーラップするように前記ニューラルネットワーク演算を実行してイントラ－レイヤパイプライニングを実現することができる。

前記プロセッサは、前記ニューラルネットワーク演算の実行中に前記暗号処理フロント－エンドプロセッサが前記暗号化入力データの復号化演算を行うようにして、前記ニューラルネットワーク演算を途切れることなく行うことができる。

前記プロセッサは、前記イントラ－レイヤパイプライニングのために、データ復号化ステップ、演算ステップ、及びデータ暗号化ステップに細分化して、前記ニューラルネットワーク演算を途切れることなく行うことができる。

【００１１】

上記目的を達成するためになされた本発明の他の態様によるトラスト環境ベースの人工知能装置は、暗号化入力データを送信する第１類型のメモリと、トラスト信頼空間で動作し、前記暗号化入力データの人工知能演算を行うトラスト人工知能処理部と、を備え、前記トラスト人工知能処理部は、前記暗号化入力データの復号化を介して復号化入力データを生成する暗号処理フロント－エンドプロセッサと、前記復号化入力データ及び非暗号化出力データに対するバッファを提供する第２類型のメモリと、前記復号化入力データに基づいてニューラルネットワーク演算を行って前記非暗号化出力データを生成するプロセッサと、を含む。

【００１２】

前記プロセッサは、直接畳み込みベースのニューラルネットワーク演算を行って前記第１類型のメモリのアクセス数を減らすことができる。

前記プロセッサは、前記第２類型のメモリに復号化入力アクティベーションを固定的に格納し、復号化フィルタ及び非暗号出力アクティベーションを循環キュー方式で格納することができる。

前記プロセッサは、前記暗号処理フロント－エンドプロセッサの暗号化及び復号化演算の実行にオーバーラップするように前記ニューラルネットワーク演算を実行してイントラ－レイヤパイプライニングを実現することができる。

【発明の効果】

【００１３】

本明細書で開示する技術は、次の効果を有する。但し、特定の実施形態が次の効果を全

て含まなければならないとか、又は次の効果のみを含まなければならないという意味ではないため、開示する技術の権利範囲はこれによって制限されるものと理解してはならない。

【００１４】

本発明のトラスト環境ベースの人工知能装置によれば、トラスト（Ｔｒｕｓｔ）環境で人工ニューラルネットワークの実行を安全に加速化することができる。

【００１５】

また、トラスト信頼空間でデータの暗号化を行って物理的攻撃からセキュリティを強化し、直接畳み込みベースのニューラルネットワーク演算を行ってメモリのアクセス数を減らすことができる。

【００１６】

更に、暗号ハードウェアでオフロードを行ってニューラルネットワークの実行作業が制限されたプロセッサリソースを活用することができ、イントラ－レイヤパイプライニングを介してデータの暗号化及び復号化にオーバーラップして人工ニューラルネットワークの実行時間を短縮することができる。

【図面の簡単な説明】

【００１７】

【図１】本発明によるトラスト環境ベースの人工知能装置を説明する図である。

【図２】図１の人工知能装置でトラスト環境ベースの人工知能演算の実行方法を説明する図である。

【図３】トラスト実行環境を説明する図である。

【図４ａ】従来と比べた本発明によるＤＮＮフレームワークの作業モデルを説明する図である。

【図４ｂ】従来と比べた本発明によるＤＮＮフレームワークの作業モデルを説明する図である。

【図５】従来のＤＮＮフレームワークの実行速度、帯域幅、及び復号化のスループットをそれぞれ示す図である。

【図６】従来のＤＮＮフレームワークの実行速度、帯域幅、及び復号化のスループットをそれぞれ示す図である。

【図７】従来のＤＮＮフレームワークの実行速度、帯域幅、及び復号化のスループットをそれぞれ示す図である。

【図８ａ】ニューラルネットワーク演算の実行に使用される畳み込みを説明する図である。

【図８ｂ】ニューラルネットワーク演算の実行に使用される畳み込みを説明する図である。

【図９ａ】ＤＮＮに優しいＳＲＡＭ管理過程を説明する図である。

【図９ｂ】ＤＮＮに優しいＳＲＡＭ管理過程を説明する図である。

【図１０】ＣＰＵと暗号ハードウェアとの間のスループットを示す図である。

【図１１】暗号ハードウェアのオフロードを説明する図である。

【図１２】ニューラルネットワーク演算の実行過程でイントラ－レイヤパイプライニングの実現を説明する図である。

【図１３】ニューラルネットワーク演算の実行過程でイントラ－レイヤパイプライニングの実現を説明する図である。

【図１４】本発明のソフトウェアの構成例を説明する図である。

【図１５】本発明による実験結果を説明する図である。

【図１６】本発明による実験結果を説明する図である。

【図１７】本発明による実験結果を説明する図である。

【発明を実施するための形態】

【００１８】

本発明は、下記の研究課題をもって支援を受けて出願された。

［この発明を支援した国家研究開発事業］

［課題固有番号］　１７１１１９３９８６

［課題番号］　２０２０－０－０１３６１－００４

［省庁名］　科学技術情報通信部
［課題管理（専門）機関名］　情報通信企画評価院
［研究事業名］　情報通信放送革新人材養成
［研究課題名］　人工知能大学院支援（延世大学校）
［寄与率］　１／２
［課題実行機関名］　延世大学校産学協力団
［研究期間］　２０２３．０１．０１～２０２３．１２．３１
［この発明を支援した国家研究開発事業］
［課題固有番号］　１４１５１７７６５９
［課題番号］　Ｐ００１６１５０
［省庁名］　産業通商資源部
［課題管理（専門）機関名］　韓国産業技術振興院
［研究事業名］　産業技術国際協力（Ｒ＆Ｄ）
［研究課題名］　スマートＩｏＴネットワークセキュリティのための攻撃探知及び防御技術開発
［寄与率］　１／２
［課題実行機関名］　韓国電子技術研究院
［研究期間］　２０２１．１２．０１～２０２２．１１．３０

【００１９】

本発明に関する説明は、構造的又は機能的説明のための実施形態に過ぎないため、本発明の権利範囲は、本明細書に説明している実施形態によって制限されるものと解釈してはならない。即ち、実施形態は、様々な変更が可能であり、種々の形態を有することができるため、本発明の権利範囲は、技術的思想を実現することができる均等物を含むものと理解しなければならない。また、本発明で提示している目的又は効果は、特定の実施形態がこれを全て含むべきであるとか、そのような効果だけを含むべきであるという意味ではないため、本発明の権利範囲は、これによって制限されるものと理解してはならない。

【００２０】

一方、本明細書で述べる用語の意味は、次のように理解されなければならない。

【００２１】

「第１」、「第２」などの用語は、１つの構成要素を他の構成要素から区別するためのものであって、これらの用語によって権利範囲が限定されてはならない。例えば、第１構成要素は第２構成要素と名付けられてもよく、同様に第２構成要素も第１構成要素と名付けられてもよい。

【００２２】

ある構成要素が他の構成要素に「連結されて」いると言及する場合には、その他の構成要素に直接連結されることもあるが、中間に他の構成要素が存在することもあると理解しなければならない。それに対し、ある構成要素が他の構成要素に「直接連結されて」いると言及する場合には、中間に他の構成要素が存在しないものと理解しなければならない。一方、構成要素間の関係を説明する他の表現、即ち「～間に」と「すぐ～間に」、又は「～に隣り合う」と「～に直接隣り合う」なども同様に解釈されなければならない。

【００２３】

単数の表現は、文脈上明白に異なる意味ではない限り、複数の表現を含むものと理解しなければならず、「含む」又は「有する」などの用語は、実施される特徴、数字、ステップ、動作、構成要素、部分品、又はこれらを組み合わせたものが存在することを指定しようとするものであり、１つ又はそれ以上の他の特徴や数字、ステップ、動作、構成要素、部分品、又はこれらを組み合わせたものの存在又は付加可能性を予め排除しないものと理解しなければならない。

【００２４】

各ステップにおいて、識別符号（例えば、ａ、ｂ、ｃなど）は、説明の便宜のために使用されるものであって、識別符号は、各ステップの順序を説明するものではなく、各ステ

ップは、文脈上明白に特定の順序を記載しない限り、明記された順序と異なって生じ得る。即ち、各ステップは、明記された順序と同一に生じることもあり、実質的に同時に行われることもあり、反対の順序で行われることもある。

【００２５】

本発明は、コンピュータ読み取り可能な記録媒体にコンピュータ読み取り可能なコードとして実現され、コンピュータ読み取り可能な記録媒体は、コンピュータシステムによって読み取られるデータが格納される全ての種類の記録装置を含む。コンピュータ読み取り可能な記録媒体の例としては、ＲＯＭ、ＲＡＭ、ＣＤ－ＲＯＭ、磁気テープ、フロッピー（登録商標）ディスク、光データ格納装置などがある。また、コンピュータ読み取り可能な記録媒体は、ネットワークで連結されたコンピュータシステムに分散されて、分散方式でコンピュータ読み取り可能なコードが格納され、実行され得る。

【００２６】

ここで使用する全ての用語は、異なって定義されない限り、本発明が属する分野における通常の知識を有する者により一般的に理解されるものと同じ意味を有する。一般的に使用される辞書に定義されている用語は、関連技術の文脈上有する意味と一致するものと解釈されなければならず、本明細書において明白に定義しない限り、理想的であるか又は過度に形式的な意味を有するものと解釈することはできない。

【００２７】

以下、本発明を実施するための形態の具体例を、図面を参照しながら詳細に説明する。

【００２８】

図１は、本発明によるトラスト環境ベースの人工知能装置を説明する図である。

【００２９】

図１を参照すると、トラスト環境ベースの人工知能装置１００は、モバイル又は組み込み機器で人工ニューラルネットワークの実行に対してセキュリティ強化及び加速化のためのフレームワークであり、第１類型のメモリ１１０及びトラスト人工知能処理部１３０を含んで実現される。

【００３０】

第１類型のメモリ１１０は、暗号化入力データを送信し、暗号化出力データを受信する。ここで、第１類型のメモリ１１０はＤＲＡＭ（Ｄｙｎａｍｉｃ　Ｒａｎｄｏｍ　Ａｃｃｅｓｓ　Ｍｅｍｏｒｙ）で構成されるが、必ずしもこれに限定されるものではない。

【００３１】

トラスト人工知能処理部１３０は、トラスト信頼空間で動作し、暗号化入力データ及び暗号化出力データの人工知能演算を行う。このために、トラスト人工知能処理部１３０は、暗号処理フロント－エンドプロセッサ１３１、第２類型のメモリ１３３、及びプロセッサ１３５を含む。

【００３２】

暗号処理フロント－エンドプロセッサ１３１は、暗号ハードウェア（Ｃｒｙｔｏｇｒａｐｈｉｃ　Ｈａｒｄｗａｒｅ）として暗号化入力データの復号化を介して復号化入力データを生成し、暗号化出力データを生成するために非暗号化出力データの暗号化を行う。暗号処理フロント－エンドプロセッサ１３１は、第１類型のメモリ１１０から、暗号化入力データである暗号化入力アクティベーション及び暗号化フィルタの入力を受けて、第２類型のメモリ１３３に復号化入力アクティベーション及び復号化フィルタを格納する。暗号処理フロント－エンドプロセッサ１３１は、人工知能演算の過程でプロセッサ１３５によるオン－デマンド（Ｏｎ－Ｄｅｍａｎｄ）要求を受信し、第１類型のメモリ１１０にアクセスして暗号化入力データを取り込む。

【００３３】

第２類型のメモリ１３３は、復号化入力データ及び非暗号化出力データに対するバッファを提供する。ここで、第２類型のメモリ１３３は、第１類型のメモリ１１０よりも相対的に速い動作速度及び少ない格納容量を有する。例えば、第１類型のメモリ１１０がＤＲＡＭで構成された場合、第２類型のメモリ１３３はＳＲＡＭ（Ｓｔａｔｉｃ　Ｒａｎｄｏｍ

Ａｃｃｅｓｓ　Ｍｅｍｏｒｙ）で構成される。第１類型のメモリ１１０と第２類型のメモリ１３３との間でデータを送受信するときに、暗号処理フロント－エンドプロセッサ１３１でデータを暗号化及び復号化して、正確性及びセキュリティ性を保証する。

【００３４】

プロセッサ１３５は復号化入力データに基づいてニューラルネットワーク演算を行って非暗号化出力データを生成する。プロセッサ１３５は直接畳み込みベースのニューラルネットワーク演算を行って第１類型のメモリ１１０のアクセス数を減らす。即ち、プロセッサ１３５は、直接畳み込みを使用してニューラルネットワーク演算の作業セットサイズを最小化し、減少した作業セットサイズにより、ニューラルネットワーク演算を行っている間、第２類型のメモリ１３３の動作速度よりも相対的に遅い第１類型のメモリ１１０に対するアクセス及びデータ暗号化及び復号化の呼び出しを減らす。

【００３５】

プロセッサ１３５は、第２類型のメモリ１３３に復号化入力アクティベーションを固定的に格納し、復号化フィルタ及び非暗号出力アクティベーションを循環キュー方式で格納する。プロセッサ１３５は、ニューラルネットワーク演算の実行で畳み込みレイヤの入力アクティベーション、フィルタ、及び出力アクティベーションのアクセスパターンに基づいて、第２類型のメモリ１３３を効率的に管理する。プロセッサ１３５は、入力アクティベーションの場合、データの再使用を増加させる傾向があるため固定的に格納し、フィルタの場合、出力チャンネルを生成するときに同じフィルタを複数回使用するが、サイズが入力アクティベーションよりもはるかに小さく、１つの出力チャンネルのみが生成するのに必要であるため、循環キュー（Ｃｉｒｃｕｌａｒ　Ｑｕｅｕｅ）方式で格納し、出力アクティベーションの場合、空間的集約性の高い１回の書き込みデータであるため、循環キュー方式で格納する。

【００３６】

プロセッサ１３５は暗号処理フロント－エンドプロセッサ１３１の割り込み駆動のオフロード（ｉｎｔｅｒｒｕｐｔ　ｄｒｉｖｅｎ　ｏｆｆｌｏａｄｉｎｇ）を介して第１類型のメモリ１１０及び第２類型のメモリ１３１とデータ送受信を行う。また、プロセッサ１３５は、ＤＭＡ（Ｄｉｒｅｃｔ　Ｍｅｍｏｒｙ　Ａｃｃｅｓｓ）コントローラのＤＭＡ駆動のオフロードを介して、暗号処理フロント－エンドプロセッサ１３１並びに第１類型のメモ１１０リ及び第２類型のメモリ１３３とデータ送受信を行う。プロセッサ１３５は割り込み駆動のオフロード及びＤＭＡ駆動のオフロードの中から選択される。割り込み駆動のオフロードは待ち時間が重要な作業を行うのに適しており、ＤＭＡ駆動のオフロードは一度により多くの量のデータを処理するのに適している。ここで、両オフロードのメカニズムは、ニューラルネットワーク演算を加速化する。

【００３７】

プロセッサ１３５は、暗号処理フロント－エンドプロセッサ１３１の暗号化及び復号化演算の実行にオーバーラップするようにニューラルネットワーク演算を実行して、イントラ－レイヤパイプライニングを実現する。プロセッサ１３５は、ニューラルネットワーク演算の実行中に暗号処理フロント－エンドプロセッサ１３１が暗号化入力データの復号化演算を行うようにして、ニューラルネットワーク演算を途切れることなく行う。プロセッサ１３５は、イントラ－レイヤパイプライニングのために、データ復号化ステップ、演算ステップ、及びデータ暗号化ステップに細分化して、ニューラルネットワーク演算を途切れることなく行う。プロセッサ１３５は、イントラ－レイヤパイプライニングを介して、データ復号化ステップ、演算ステップ、及びデータ暗号化ステップを並列化する。

【００３８】

図２は、図１の人工知能装置でトラスト環境ベースの人工知能演算の実行方法を説明するフローチャートである。

【００３９】

図２において、人工知能装置１００は、第１類型のメモリ１１０から暗号化入力データの入力を受けて、復号化を介して復号化入力データを生成する（ステップＳ２１０）。人

工知能装置１００は暗号処理フロント－エンドプロセッサ１３１を介して暗号化入力データに対して復号化を行う。人工知能装置１００は復号化入力データを第２類型のメモリ１３３に格納する。

【００４０】

人工知能装置１００は、復号化入力データに基づいてニューラルネットワーク演算を行って非暗号化出力データを生成する（ステップＳ２３０）。ここで、人工知能装置１００はプロセッサ１３５を介して直接畳み込みベースのニューラルネットワーク演算を行って暗号処理フロント－エンドプロセッサ１３１が第１類型のメモリ１１０にアクセスして暗号化入力データを取り込むアクセス数を減らす。人工知能装置１００は、非暗号化出力データを第２類型のメモリ１３３に格納する。

【００４１】

人工知能装置１００は、非暗号化出力データの暗号化を行って暗号化出力データを生成する（ステップＳ２５０）。人工知能装置１００は、暗号処理フロント－エンドプロセッサ３１０を介して非暗号化出力データに対して暗号化を行う。人工知能装置１００は暗号化出力データを第１類型のメモリ１１０に出力する。

【００４２】

人工知能装置１００は、暗号処理フロント－エンドプロセッサ１３１の暗号化及び復号化演算の実行にオーバーラップするようにプロセッサ１３５のニューラルネットワーク演算を実行して、データ復号化ステップ、演算ステップ、及びデータ暗号化ステップでイントラ－レイヤパイプライニングを実現することによって、ニューラルネットワーク演算を加速化する。

【００４３】

以下、図３～１７を参照して、本発明によるトラスト環境ベースの人工知能装置についてより詳しく説明する。

【００４４】

図３は、トラスト実行環境を説明する図である。

【００４５】

図３を参照すると、トラスト実行環境（Ｔｒｕｓｔｅｄ　Ｅｘｅｃｕｔｉｏｎ　Ｅｎｖｉｒｏｎｍｅｎｔ：以下、ＴＥＥという）は、処理、メモリ、及び格納機能を備えたセキュリティ処理環境であり、敏感な作業及びデータが当該環境を外れないように制限し、高いセキュリティを達成する。ＴＥＥ内の作業及びデータは、ＯＳ（Ｏｐｅｒａｔｉｎｇ　Ｓｙｓｔｅｍ）及びアプリケーションが実行されるリッチ実行環境（Ｒｉｃｈ　Ｅｘｅｃｕｔｉｏｎ　Ｅｎｖｉｒｏｎｍｅｎｔ：以下、ＲＥＥという）で隔離される。

【００４６】

モバイル及び組み込み装置で使用されるトラストゾーンサポートのＴＥＥを実現するために、トラストゾーンはＣＰＵが指定された時点にＴＥＥ及びＲＥＥのうちの１つでのみ独占的に作動するようにするセキュリティプロセッサモードを実現する。ＴＥＥとＲＥＥとの間の切替え及び相互作用は、ＳＭＣ（Ｓｅｃｕｒｅ　Ｍｏｎｉｔｏｒ　Ｃａｌｌｓ）で呼び出せるセキュリティモニターにより管理される。また、トラストゾーンは、ＤＲＡＭをセキュリティ及び非セキュリティ領域に分割し、ＴＥＥの敏感なデータを保護するためにＲＥＥでセキュリティ領域にアクセスすることを許容しない。

【００４７】

トラストゾーンサポートのＴＥＥで実行される既存のＤＮＮフレームワークは、１）周辺装置から入力データ（例えば、指紋センサの指紋イメージ）を受信してＲＥＥから開始される。２）ＲＥＥでＤＮＮのいくつかの初期レイヤを実行した後、３）出力アクティベーションを暗号化し、ＳＭＣを介してＴＥＥに送信する。その次に、４）ＴＥＥ内で送信されたアクティベーションを復号化し、５）復号化されたアクティベーション及び事前送信されたフィルタを使用して残りのレイヤを実行する。残りのレイヤの実行を完了すると、６）ＤＮＮが作った予測を、ＳＭＣを介してＲＥＥに返還する。

【００４８】

このような方式でＤＮＮの実行を隔離して、いくつかのセキュリティ攻撃からＤＮＮを保護する。

【００４９】

図４ａ及び４ｂは、従来と比べた発明によるＤＮＮフレームワークの作業モデルを説明する図であって、図４ａは従来のＤＮＮフレームワークであり、図４ｂは、本発明で提案するＤＮＮフレームワーク（以下、ＧｕａｒｄｉａｎＮＮという）である。

【００５０】

図４ａの従来のＤＮＮフレームワークの場合、最初にＤＮＮの入力データ及びフィルタが全て暗号化されてＤＲＡＭに格納される。フレームワークがＤＮＮのレイヤの実行を開始すると、１）暗号化された入力アクティベーション及びフィルタをＳＲＡＭにロードし、２）ＣＰＵを使用してデータを復号化する。その次に、フレームワークは、３）復号化されたデータを使用して各レイヤを実行し、出力アクティベーションをＳＲＡＭに格納する。その後、フレームワークは、４）出力アクティベーションを暗号化し、５）これをＤＲＡＭに格納する。入力アクティベーション及びフィルタは必要に応じてＳＲＡＭに置き換えられる。ＳＲＡＭがレイヤの作業集合よりも小さい場合、データスワッピング及び暗復号化がよく発生する。暗号化されたＤＲＡＭを使用してＴＥＥ内でＤＮＮを実行すると、従来のＤＮＮフレームワークがセキュリティ攻撃から敏感なユーザ及びＤＮＮデータを完全に保護できるが、遅いＤＮＮの実行で困難をきたすことができる。

【００５１】

図５乃至図７は、従来のＤＮＮフレームワークの実行速度、帯域幅、及び復号化のスループットをそれぞれ示す図である。

【００５２】

図５に示すように、図４ａの従来のＤＮＮフレームワーク（ＯＰ－ＴＥＥ　ｗｉｔｈ　Ｐａｇｅｒ）とＤＲＡＭデータを暗号化しない安全ではないＤａｒｋｎｅＴＺ作業モデルとのＤＮＮの実行速度を比較した結果であって、従来のＤＮＮフレームワークのＤＮＮの実行速度がＤａｒｋｎｅＴＺよりも顕著に遅いことが示された。従来のＤＮＮフレームワークはＤａｒｋｎｅＴＺよりもＡｌｅｘＮｅｔを実行するのに３．４２倍長くかかった。

【００５３】

従来のＤＮＮフレームワークは、組み込みＳＲＡＭの制限された容量により遅いＤＲＡＭのアクセス数が増加し、ＣＰＵに課される高いデータの暗号化及び復号化のオーバーヘッドによる性能のボトルネック現象がＤＮＮの実行速度に影響を与える。具体的に、従来のＤＮＮフレームワークは、ＴＥＥが暗号化されたＤＲＡＭデータを安全にロードし、ロードされたデータを復号化する安全なオンチップメモリとして組み込みＳＲＡＭを活用する。しかし、ＳＲＡＭの容量（数百ＫＢ）は、一般的にモバイル及び組み込み装置のオンチップＣＰＵキャッシュ（数百ＫＢ）よりも小さいため、ＳＲＡＭを安全なオンチップバッファとして使用するが、効果的なオンチップメモリのサイズを一桁に下げる。これにより、ＤＮＮを実行するときに、遅いＤＲＡＭのアクセス数が増加して、ＤＮＮの実行速度が遅くなる。

【００５４】

図６に示すように、作業セットサイズが多様なセキュリティ内蔵型ＳＲＡＭとオフチップＤＲＡＭとの帯域幅を比較した結果、ＳＲＡＭが多様な作業セットサイズに対して持続的にＤＲＡＭよりも高い帯域幅を提供することが示された。これは、増加したＤＲＡＭアクセスがＤＮＮの実行速度に否定的な影響を与えることを意味する。結果は、またより高い帯域幅を達成するためにシーケンシャルアクセスが任意のアクセスよりも好まれなければならないということを示唆する。結果として、遅いＤＲＡＭアクセスがＤＮＮの実行速度に及ぼす否定的な影響を最小化するためには、速いＳＲＡＭデータの再使用を最大化して、ＤＮＮの実行のメモリアクセスをシーケンシャルアクセスで構成しなければならない。

【００５５】

組み込みＳＲＡＭと暗号化されたＤＲＡＭとの間でデータを交換するときに、機能的正確性及び高いセキュリティ性を保証するために、ＣＰＵでデータを暗号化及び復号化しな

ければならない。しかし、ＣＰＵベースの暗復号化は速度が遅いだけでなく、制限されたＣＰＵ帯域幅のかなりの部分を消費するため、制限されたＣＰＵ帯域幅がコンピューティング集約的なＤＮＮの実行と暗号化及び復号化とで全て共有され、ＤＮＮの実行速度が遅くなる。

【００５６】

図７に示すように、ペイロードサイズが多様なＣＰＵベースのデータ復号化のスループットは、データの暗号化及び復号化が制限されたＣＰＵリソースによりも困難をきたしており、２ＫＢのペイロードで４．５１ＭＢ／ｓのスループットを達成することを示す。２ＫＢのＤＲＡＭデータにシーケンシャルアクセスするスループット（６０９．２４ＭＢ／ｓ、図６の（ａ））と比較した場合、ＣＰＵベースの２ＫＢデータの暗号化及び復号化はシーケンシャルＤＲＡＭアクセスよりも１３５．０９倍遅い。低いデータの暗号化及び復号化のスループットは、データの暗号化及び復号化のオーバーヘッドが相当であり、従来のＤＮＮフレームワークが速いＤＮＮの実行を達成することは難しいということを示す。結果として、従来のＤＮＮフレームワークは制限されたＣＰＵリソースに課される高いＤＲＡＭデータの暗号化及び復号化のオーバーヘッドによりかなりの困難をきたしており、速く安全なＤＮＮの実行のためには、オーバーヘッドを解決しなければならない。

【００５７】

従って、本発明は、従来のＤＮＮフレームワークの遅いＤＮＮの実行問題を解決し、速く安全なモバイル及び組み込み装置用ＤＮＮフレームワークであるＧｕａｒｄｉａＮＮを提案する。本発明で提案したＧｕａｒｄｉａＮＮは、次のような特徴を有する。

【００５８】

－直接畳み込みを介してＤＮＮ実行の作業セットサイズを最小化することができる。作業セットサイズを減少させて、ＤＮＮの実行中の、遅いＤＲＡＭアクセス及びデータの暗号化及び復号化要求を減らすことができる。

【００５９】

－ＤＮＮに優しいＳＲＡＭ管理を使用し、畳み込みレイヤのデータの再使用を最大化することができる。入力アクティベーションをＳＲＡＭに固定し、フィルタ及び出力アクティベーションを循環キュー方式で格納して、ＳＲＡＭを効率的に管理することができる。

【００６０】

－暗号ハードウェアでデータの暗号化及び復号化をオフロードすることができる。データの暗号化及び復号化をオフロードすると、制限されたＣＰＵリソースをＤＮＮ専用に使用することができる。

【００６１】

－ＤＮＮレイヤとデータの暗号化及び復号化作業とをオーバーラップして、ＤＮＮの実行を更に加速化することができる。これは、ＣＰＵと暗号ハードウェアとがそれぞれＤＮＮとデータの暗号化及び復号化作業とを同時に行うことができるため可能である。

【００６２】

図４ｂを見ると、本発明で提案したＤＮＮフレームワークの作業モデルは、ＤＮＮレイヤを実行するときに、１）モバイル及び組み込み装置で使用される暗号ハードウェア（図１の暗号処理フロント－エンドプロセッサ）によりＤＲＡＭで暗号化された入力アクティベーション及びフィルタをロードしてデータを復号化し、復号化データをＳＲＡＭに格納する。その次に、２）ＣＰＵ（図１のプロセッサ）は、ＳＲＡＭデータを使用してＤＮＮレイヤの作業を行い、出力アクティベーションをＳＲＡＭ（図１の第２類型のメモリ）に格納する。その後、３）暗号ハードウェアは、ＳＲＡＭに格納された出力アクティベーションを暗号化し、暗号化された出力アクティベーションをＤＲＡＭ（図１の第１類型のメモリ）に格納する。

【００６３】

図４ａの従来のＤＮＮフレームワークの作業モデルと比較すると、直接畳み込み及びＤＮＮに優しいＳＲＡＭ管理を使用し、遅いＤＲＡＭアクセス及び暗号ハードウェアを使用して、ＣＰＵに課される高いデータの暗号化及び復号化のオーバーヘッドを大幅に減らす

ことができる。本発明で提案した主要な特徴がＤＮＮの実行に与える影響は下記表１にまとめることができる。

【００６４】

【表１】

Key Idea	Beneficial to DNNs Having
Direct Convolutions	Large filter sizes, high input channel counts, small stride sizes
DNN-Friendly SRAM Mgmt.	High output channel counts
Cryptographic Hardware	Large memory footprint sizes
Intra-Layer Pipelining	Balanced compute & encryption/decryption

【００６５】

本発明は、直接畳み込みを使用して畳み込みレイヤの作業セットサイズを減らし、ＤＮＮに優しいＳＲＡＭ管理を使用してＳＲＡＭデータの再使用を最大化し、ＤＮＮの実行中の遅いＤＲＡＭのアクセスを大幅に減らすことができる。

【００６６】

図８ａ及び８ｂは、ニューラルネットワーク演算の実行に使用される畳み込みを説明する図であって、図８ａはｉｍ２ｃｏｌ（Ｉｍａｇｅ　ｔｏ　Ｃｏｌｕｍｎ）の畳み込みであり、図８ｂは直接畳み込みである。

【００６７】

従来のモバイル及び組み込みＤＮＮフレームワーク（例えば、ＤａｒｋｎｅＴＺ、ＴｅｎｓｏｒＦｌｏｗ　Ｌｉｔｅ）は、ｉｍ２ｃｏｌ（Ｉｍａｇｅ　ｔｏ　Ｃｏｌｕｍｎ）畳み込みを使用して、時間が多くかかる畳み込みレイヤを実行する。ｉｍ２ｃｏｌは、多次元データを行列に変換して、行列演算を行うようにする関数をいう。多次元データの畳み込みは、ｉｍ２ｃｏｌを介して行列に変換されたデータの内積と同じである。ｉｍ２ｃｏｌの畳み込みは、図８ａのように各パッチ（即ち、要素がフィルタの要素にマッピングされる入力アクティベーションの集合）を２次元行列に平面化し、平面化されたパッチとフィルタとの間の行列乗算を行って速い畳み込みレイヤの実行を達成する。しかし、パッチを併合すると、併合されたパッチを格納するための追加のバッファを割り当てなければならないため、割り込みレイヤの作業セットサイズが大幅に増加する。ＳＲＡＭの制限された容量（最大数百ＫＢ）により作業セットサイズが増加すると、ＤＮＮの実行速度が大幅に遅くなる遅いＤＲＡＭのアクセスが多く発生する。従って、速いＤＮＮの実行を達成するために、畳み込みレイヤの作業セットサイズを最小化しなければならない。

【００６８】

畳み込みレイヤの作業セットサイズを最小化するために、本発明ではｉｍ２ｃｏｌ畳み込みの代わりに直接畳み込みを使用する。直接畳み込みは、入力に畳み込みレイヤのフィルタを相関演算して、フィルタをスライディングウィンドウ方式で全ての領域で相関値を計算した結果値が出力となる。直接畳み込みは、図８ｂのように必要な入力アクティベーションを要求する際に取り込むため、作業セットサイズを大きくしない。減少した作業セットサイズ以外にも、直接畳み込みは、入力アクティベーションの固有の時間及び空間的地域性により、ＳＲＡＭデータの再使用を増加させる傾向がある。直接畳み込みが入力アクティベーションにかけてフィルタを空間的にスライドして出力アクティベーションのチャンネルを生成するため、最近アクセスした入力アクティベーションに隣接する入力アクティベーションが近い将来にアクセスする可能性が高い。フィルタの場合、出力チャンネルを生成するときに同じフィルタを複数回使用するため、フィルタは時間的地域性が高い。需要ページング内蔵のＳＲＡＭに結合された直接畳み込みの高い空間的及び時間的集約性は、遅いＤＲＡＭアクセスを大幅に減らすのに役立つ。

【００６９】

図９ａ及び９ｂは、ＤＮＮに優しいＳＲＡＭ管理過程を説明する図である。

【００７０】
図９ａに示すように、ＡＲＭトラストゾーン技術が適用された既存のオープンソースＴＥＥのフレームワークであるＰａｇｅｒは、デマンドページング（Ｄｅｍａｎｄ　Ｐａｇｉｎｇ）を使用してセキュリティ組み込みＳＲＡＭの制限された容量を管理する。デマンドページングは、必要に応じて暗号化されたＤＲＡＭからＳＲＡＭにデータをロードし、最も近くに使用されていないＳＲＡＭデータをＤＲＡＭにプッシュしてＳＲＡＭ空間を回収する。デマンドページングはデータの時間的地域性を効率的に活用するが、畳み込みレイヤの入力アクティベーション、フィルタ、及び出力アクティベーションの様々なアクセスパターンを認識できない。また、ＳＲＡＭのサイズは、モバイル及び組み込み装置で数百ＫＢに過ぎないため、畳み込みレイヤに必要な全てのデータをＳＲＡＭに格納することはできない。このような非効率的なＳＲＡＭ管理は、遅いＤＲＡＭのアクセス数を大幅に増加させ、速いＤＮＮの実行を妨げる。従って、ＳＲＡＭが多数の遅いＤＲＡＭアクセスを誘発しないように、畳み込みレイヤの様々なデータアクセスパターンを統合して、小さいＳＲＡＭを効率的に管理しなければならない。
【００７１】
図９ｂに示すように、本発明は、ＳＲＡＭデータの再使用を最大化するために、ＤＮＮに優しいＳＲＡＭ管理を実現する。ここで、ＤＮＮに優しいＳＲＡＭ管理は、畳み込みレイヤの入力アクティベーション、フィルタ、及び出力アクティベーションの様々なアクセスパターンを活用する。本発明は、畳み込みレイヤの実行全体で繰り返しアクセスするときに、先ず全ての入力アクティベーションをＳＲＡＭに固定する。出力チャンネルを生成するためにアクセスされる入力アクティベーションは、他の全ての出力チャンネルに対して再度アクセスされなければならない。その次に、残りのＳＲＡＭ空間を２つの循環キューで構成する。１つはフィルタ用であり、もう１つは出力アクティベーション用である。フィルタは入力アクティベーションのようによくアクセスするデータである。しかし、サイズが入力アクティベーションよりもはるかに小さく、１つの出力チャンネルのみを生成するのにフィルタが必要であるため、フィルタを循環キューに格納するだけでも高い時間的地域性を十分活用することができる。出力アクティベーションは、空間的集約性の高い１回の書き込みデータであるため、循環キューに適している。ＤＮＮに優しい方式でＳＲＡＭを管理することによって、ＳＲＡＭデータの再使用を最大化して遅いＤＲＡＭアクセスを最小化し、ＤＮＮの実行を加速化することができる。
【００７２】
本発明は、暗号ハードウェアを使用してデータの暗号化及び復号化を行うようにして、制限されたＣＰＵリソースをＤＮＮの実行に完全に使用することができる。これにより、ＣＰＵリソースをＤＮＮの実行に完全に専用するだけでなく、暗号ハードウェアの高性能を活用して速いＤＮＮの実行を達成することができる。データの暗号化及び復号化は、ＳＲＡＭが暗号化されたＤＲＡＭでデータをロード及び格納して、制限されたＣＰＵリソースのかなりの量を消費する度に発生する。データの暗号化及び復号化の高いオーバーヘッドを克服するために、暗号ハードウェアを使用して暗号化及び復号化作業を行う。オーバーヘッドの高いデータの暗号化及び復号化を暗号ハードウェアにオフロードし、制限されたＣＰＵリソースを完全に専用して、ＤＮＮの実行を加速化することができる（図１０参照）。
【００７３】
図１１は、暗号ハードウェアのオフロードを説明する図であって、（ａ）は割り込みベースのオフロードであり、（ｂ）はＤＭＡ（Ｄｉｒｅｃｔ　Ｍｅｍｏｒｙ　Ａｃｃｅｓｓ）駆動のオフロードである。
【００７４】
本発明は、図１１の（ａ）割り込みベースのオフロード、及び（ｂ）ＤＭＡ駆動のオフロードの２つのハードウェアのオフロードメカニズムから選択される。２つのオフロードメカニズムは、互いに異なる周辺装置の使用シナリオで加速化される。割り込みベースのオフロードは、周辺装置が与えられた作業処理を完了するとすぐに割り込みを生成するた

め、待ち時間が重要な作業を周辺装置で行うのに適している。これに対し、ＤＭＡ駆動のオフロードは、一度により多くの量のデータを処理するのに適している。ＤＭＡは、ＣＰＵが介入することなくデータを処理するための全てのデータ転送及び周辺の呼び出しを処理し、全てのデータが処理された後に割り込みを発生させる。ＤＮＮに優しいＳＲＡＭ管理は、ＳＲＡＭとＤＲＡＭとの間で大量のデータ転送（一般に数十ＫＢ）を要求するため、割り込みベースのオフロードよりも大きなペイロードサイズでより高いスループットを達成するＤＭＡベースのオフロードが有利なことがある。

【００７５】

暗号ハードウェアを使用すると、制限されたＣＰＵリソースをＤＮＮの実行に完全に専用し、ＤＮＮの実行を加速化することができ、更なる性能の最適化が可能である。ＣＰＵ及び暗号ハードウェアでそれぞれ実行される重複ＤＮＮの実行及びデータの暗号化及び復号化である。畳み込みレイヤの１つの属性は、互いに異なる出力チャネルを生成する作業が互いに独立していることである。畳み込みレイヤは１つのフィルタを使用して１つの出力チャンネルを生成し、入力アクティベーションは全ての出力チャンネル間で読み取り専用データとして共有される。この属性はプーリングレイヤにも適用される。プーリングレイヤの各出力チャンネルは、当該入力チャンネルのみを要求するため、出力チャンネルの作業をデータ並列にする。これに基づいて、畳み込みレイヤの大量の出力チャンネルの実行をデータの復号化ステップ、演算ステップ、及びデータの暗号化ステップの３つのパイプラインステップに分割する。その次に、出力チャンネルの互いに異なるバルクの３つのステップをパイプラインして、より速いＤＮＮの実行を達成する。これをイントラ－レイヤパイプライニングという。これは、畳み込みレイヤの様々な出力チャンネル作業をパイプラインで接続するためである。

【００７６】

図１２は、ニューラルネットワーク演算の実行過程でイントラ－レイヤパイプライニングの実現を説明する図であり、２５６個の出力チャンネル（ＳＲＡＭの制限された容量により大量の３２個の出力チャンネル含む）を生成し、ＡｌｅｘＮｅｔの実行待ち時間に最も大きく寄与するＡｌｅｘＮｅｔの６番目のレイヤにイントラ－レイヤパイプライニングを適用する場合の利点を示す。

【００７７】

図１２において、データの復号化ステップは、先ず大量の出力チャンネルを生成するのに必要な入力アクティベーション及び／又はフィルタを復号化してＳＲＡＭにロードする。その次に、演算ステップでは入力アクティベーション及びフィルタを使用して大量の出力チャンネルを演算する。その後、データの暗号化ステップで大量の出力チャンネルを暗号化してＤＲＡＭに格納する。イントラ－レイヤパイプライニングがない場合、（ａ）のように３つのステップが直列化され、レイヤを実行するのに５７ｍｓがかかる。これに対し、出力チャンネルの互いに異なるバルクの３つのステップをパイプラインするイントラ－レイヤパイプライニングは、（ｂ）のように３つのステップが並列化されてレイヤを実行するのに（ａ）の場合よりも２４．６％更に速い４３ｍｓしかかからない。この例は、イントラ－レイヤパイプライニングがより速いＤＮＮの実行を達成することができることを示す。図１３は、畳み込みレイヤにイントラ－レイヤパイプライニングを適用するための擬似コードを示すアルゴリズムである。

【００７８】

しかし、畳み込みレイヤにイントラ－レイヤパイプライニングを適用すると、パイプラインを使用しないレイヤの実行よりもＳＲＡＭの容量がより多く消費される。イントラ－レイヤパイプライニングの機能的正確性を保証するためには、より大きな容量が必要である。大量の出力チャンネルの演算ステップと次の大量の出力チャンネルのデータ復号化ステップとをオーバーラップするためには、２つの大量に対するＳＲＡＭバッファが同時に割り当てられなければならない。より大きなＳＲＡＭバッファが必要な場合、レイヤの出力チャンネルはより多くのバルクにグループ化されるため、ＤＮＮの実行速度が遅くなる可能性がある。しかし、より大きなＳＲＡＭバッファが必要であるにも拘らず、イントラ

－レイヤパイプライニングの性能の利点は、バルク当たりの出力チャンネル数が少なく、潜在的な性能低下を超えることにある。従って、基本的に畳み込み及びプーリングレイヤに対するイントラ－レイヤパイプライニングをアクティベーションする。しかし、非常に小さな組み込みＳＲＡＭが装着されたモバイル及び組み込み装置の場合、潜在的な性能低下を防止するために、イントラ－レイヤパイプライニングを非アクティベーションすることができる。

【００７９】

図１４は、セキュリティ組み込みＳＲＡＭ及び暗号ハードウェアが装着されたモバイル及び組み込み装置上に本発明を実現するのに必要なソフトウェアの構成例を説明する図である。

【００８０】

図１４において、最も広く使用されるＴＥＥ実現のうちの１つであるオープンソースＴｒｕｓｔＺｏｎｅベースの実現であるＯＰ－ＴＥＥ上に本発明を実現しているが、必ずしもこれに限定されるものではなく、全てのＴｒｕｓｔＺｏｎｅベースのＴＥＥ実現上に実現され得る。本発明の核心には、ＴＥＥ内の与えられた入力に対する予測を生成するためにＤＮＮのレイヤを実行する信頼できるアプリケーションであるＧｕａｒｄｉａＮＮランタイムがある。ＤＮＮの実行はＲＥＥから開始される。先ず、ＲＥＥは説明（例えば、レイヤ数、レイヤ当たりの入力及び出力アクティベーションサイズ）、暗号化入力データ、及びＤＮＮのフィルタをＴＥＥメモリに送信し、ＧｕａｒｄｉａＮＮランタイムを呼び出す。その次に、ＧｕａｒｄｉａＮＮランタイムはそれぞれ異なる類型のＤＮＮのレイヤを実現するヘルパー関数を使用してＤＮＮの各レイヤを実行する。その後、ＧｕａｒｄｉａＮＮランタイムはＤＮＮを実行して作られた予測をＲＥＥに再度返還する。

【００８１】

直接畳み込みは、ＧｕａｒｄｉａＮＮランタイム内でのみ実現することができ、ＧｕａｒｄｉａＮＮランタイムが信頼できるＯＳ（例えば、ＯＰ－ＴＥＥ）と相互作用してＳＲＡＭを割り当てて、暗号ハードウェア及びＤＭＡを使用しなければならない。ＤＮＮに優しいＳＲＡＭ管理及びイントラ－レイヤパイプライニングを実現するためには、ＧｕａｒｄｉａＮＮランタイムがＤＲＡＭに置き換えられないバッファをＳＲＡＭに割り当てなければならない。このために、ＴＥＥメモリの割り当てのためのＴＥＥ　Ｉｎｔｅｒｎａｌ　Ｃｏｒｅ　ＡＰＩ関数であるＴＥＥ＿Ｍａｌｌｏｃ（）を拡張し、ｉｓＳＲＡＭという追加入力因数を使用する。ｉｓＳＲＡＭの値がｔｒｕｅである場合、信頼できるＯＳはＳＲＡＭバッファを割り当てて、バッファがＤＲＡＭにスワップアウトされることを防止する。また、信頼できるＯＳ（例えば、Ｐａｇｅｒ）のメモリ管理を拡張し、ｉｓＳＲＡＭをＤＲＡＭにｔｒｕｅに設定した状態でＴＥＥ＿Ｍａｌｌｏｃ（）を呼び出し、割り当てられたＳＲＡＭバッファを置き換えることを防止する。ｉｓＳＲＡＭの基本値はｆａｌｓｅに設定され、ｉｓＳＲＡＭを認識できない既存の信頼できるアプリケーションの機能的正確性を保証する。例えば、ＴＥＥ＿Ｍａｌｌｏｃ（１０２４、ｈｉｎｔ、ｔｒｕｅ）を呼び出すと、ＤＲＡＭに除去されない１ＫＢ　ＳＲＡＭバッファが割り当てられる。ここで、ヒントはバッファの特性に関するいくつかのヒントを提供する（例えば、０で満たされる）。

【００８２】

ＤＭＡ駆動のデータ暗号化及び復号化のオフロードの場合、ＧｕａｒｄｉａＮＮランタイムは信頼できるＯＳのＤＭＡ装置ドライバで定義されたユーザ指定システム呼び出しを呼び出す。ＧｕａｒｄｉａＮＮの実現は、信頼できるＯＳを拡張して、ＥｎｃｒｙｐｔＤａｔａ（）及びＤｅｃｒｙｐｔＤａｔａ（）という２つのユーザ指定システム呼び出しを提供する。ＥｎｃｒｙｐｔＤａｔａ（）システム呼び出しは、暗号コンテキスト（暗号類型、キーサイズ等含む）、ＳＲＡＭ開始アドレス、ＤＲＡＭ開始アドレス、及びデータサイズを入力として使用する。それから、システム呼び出しは、ＣＰＵキャッシュをフラッシュしてダーティキャッシュラインをＳＲＡＭから除去し、ＳＲＡＭ開始アドレスで暗号化されていないＳＲＡＭデータを読み取り、暗号化コンテキスト及び暗号ハードウェアを

使用してデータを暗号化し、暗号化されたデータをＤＲＡＭ開始アドレスでＤＲＡＭに格納する。類似の方式で、ＤｅｃｒｙｐｔＤａｔａ（）システム呼び出しは、暗号コンテキスト、ＤＲＡＭ開始アドレス、ＳＲＡＭ開始アドレス、及びデータサイズの四つの因数を入力として使用する。その次に、ＥｎｃｒｙｐｔＤａｔａ（）システム呼び出しと類似する手順によって暗号化されたＤＲＡＭデータを読み取り、データを復号化し、復号化されたデータをＳＲＡＭに格納する。

【００８３】

拡張されたＴＥＥ＿Ｍａｌｌｏｃ（）ＡＰＩ機能及び既存のＧｌｏｂａｌＰｌａｔｆｏｒｍ ＡＰＩと共に２つのユーザ指定システム呼び出しを使用すると、本発明をモバイル及び組み込み装置上に充実に実現することができる。

【００８４】

≪評価（ＥＶＡＬＵＡＴＩＯＮ）≫

【００８５】

＜実験セットアップ（Ｅｘｐｅｒｉｍｅｎｔａｌ Ｓｅｔｕｐ）＞

【００８６】

速く安全なＤＮＮの実行に対するＧｕａｒｄｉａＮＮの効果を評価するために、ＳＴＭ３２ＭＰ１５７Ｃ－ＤＫ２開発ボード上にＧｕａｒｄｉａＮＮをプロトタイプにして、ＤＮＮの実行速度及びエネルギー消費を基本セキュリティＤＮＮフレームワークと比較した。開発ボードは、オープンソーストラストゾーン（ＴｒｕｓｔＺｏｎｅ）ベースのＴＥＥ実現であるＯＰ－ＴＥＥで公式的にサポートし、最新の組み込み装置の一般的なハードウェアの構成を反映した。デュアルコアＡＲＭ Ｃｏｒｔｅｘ－Ａ７ ＣＰＵ、２５６ＫＢのセキュリティ組み込みＳＲＡＭ、暗号ハードウェア、及び５１２ＭＢ ＤＤＲ３Ｌ ＤＲＡＭで構成される。信頼できるＯＳとしてＧｌｏｂａｌＰｌａｔｆｏｒｍ ＴＥＥ Ｃｌｉｅｎｔ ＡＰＩ ｖ１．１及びＩｎｔｅｒｎａｌ Ｃｏｒｅ ＡＰＩ ｖ１．０をサポートするＰａｇｅｒと共にＯＰ－ＴＥＥ ｖ３．１１．０を使用する。ＧｕａｒｄｉａＮＮ及び基本フレームワークの実現は、ＯＰ－ＴＥＥが現在単一のＴＥＥインスタンス内でマルチスレッドをサポートしないため、１つのＣＰＵコアのみ使用する。ＡＲＭ ＮＥＯＮ単一コマンドの多重データコマンドでＤａｒｋｎｅＴＺのＤＮＮレイヤの実現を拡張し、各ＤＮＮレイヤ作業間のデータ並列性を活用した。拡張ＤＮＮレイヤの実現はＧｕａｒｄｉａＮＮ及び基本フレームワークに全て適用される。信頼できるＯＳが全てのＳＲＡＭ（ＴＥＥとＲＥＥとの間の共有メモリでＯＰ－ＴＥＥが予約した４ＫＢを除外）をＧｕａｒｄｉａＮＮ及び基本フレームワークに全て割り当てると仮定する。エネルギー消費の比較のために、Ｍｏｎｓｏｏｎ ＨＶＰＭ（Ｈｉｇｈ Ｖｏｌｔａｇｅ Ｐｏｗｅｒ Ｍｏｎｉｔｏｒ）を使用して、装置全体のエネルギー消費を測定する。

【００８７】

ベンチマークとして８ビットの整数量子化を使用して量子化された８個のＤＮＮを選択し、敏感なユーザ及びＤＮＮデータに関連する５個の代表的なモバイル及び組み込みアプリケーションのドメインを取り扱う。５個のドメインは、イメージ分類、顔認識、指紋認識、視線追跡、及び感情認識である。各ドメインに対して多様なＤＮＮが存在する。しかし、ここで、実行するのに合理的に短い実行待ち時間を有する各ドメインで代表的な軽量のＤＮＮを選択する。例えば、ＳＴＭ３２ＭＰ１５７Ｃ－ＤＫ２開発ボードで基本フレームワークがＤＮＮを実行するのに１９２秒がかかったため、イメージ分類のためのＤＮＮであるＲｅｓＮｅｔ－１８をベンチマークに含めない。下記の表２には８個のＤＮＮ及びその特性が並べられている。

【００８８】

【表２】

Domain	Name	# Layers	Input Size
Image classification	AlexNet [35]	11	3x32x32
	MobileNetV1 [25]	29	3x32x32
	Tiny Darknet [63]	21	3x32x32
	VGG-7 [37]	11	3x32x32
Face recognition	DeepID [69]	9	3x31x31
Fingerprint recognition	FgptAuth [66]	11	1x128x128
Gaze tracking	GAZEL [58]	10	1x64x64
Emotion recognition	Smart Doll [11]	11	3x50x50

【００８９】

＜速いＤＮＮの実行（Ｆａｓｔ　ＤＮＮ　Ｅｘｅｃｕｔｉｏｎ）＞

【００９０】

先ず、選択された全てのＤＮＮの実行待ち時間を測定し、ＧｕａｒｄｉａＮＮのＤＮＮの実行速度を評価する。提案された技術の寄与度を分析するために、基本フレームワークから開始して、各提案された技術を漸進的に適用し、ＤＮＮの実行待ち時間を測定する。直接畳み込み及びＤＮＮに優しいＳＲＡＭ管理、暗号ハードウェア、及びイントラ－レイヤパイプラインに対する作業をすると、図１５に５個の棒で表示された計５個の構成が提供される。実験でイントラ－レイヤパイプライニングのために８個の出力チャンネルのバルクサイズを使用する。

【００９１】

図１５の（ａ）は待ち時間の実験結果を示す。殆どのＤＮＮの実行は、基本フレームワークで極めて遅く、最大１１．４秒の待ち時間が発生する。ＧｕａｒｄｉａＮＮは全てのＤＮＮで待ち時間を１秒未満に減らす。図１５の（ｂ）は、直接畳み込み及びＤＮＮに優しいＳＲＡＭ管理技術（三番目の棒）が適用された構成で正規化されたＧｕａｒｄｉａＮＮが提供する相対的速度の向上を示す。それぞれの提案された技術が効果的であることを観察することができる。直接畳み込みを適用すると、ｉｍ２ｃｏｌを使用する基本フレームワークに比べて幾何平均速度が２．５８倍向上する。ＤＮＮに優しいＳＲＡＭ管理は３．１９倍の追加の幾何平均速度の向上をもたらす。暗号化及び復号化を暗号ハードウェアでオフロードすると、１．７３倍の追加の幾何平均速度の向上が提供される。イントラ－レイヤパイプライニングを適用すると、これを更に改善し、幾何平均速度を１．０７倍高めることができる。ＧｕａｒｄｉａＮＮはベースラインに比べて１５．３倍の幾何平均速度の向上を達成する。ここで評価する８個のＤＮＮのうち、ＧｕａｒｄｉａＮＮはＳｍａｒｔ　Ｄｏｌｌを最も加速化し、ベースラインよりも３１．４倍更に速くする。このような速度向上の殆どは、ＤＮＮに優しいＳＲＡＭ管理を直接畳み込みと共に適用した結果、遅いＤＲＡＭのアクセス数が減少したためである。ＡｌｅｘＮｅｔはＧｕａｒｄｉａＮＮの暗号ハードウェア及びイントラ－レイヤパイプラインの使用で最も大きい利点を得る。レイヤ演算に比べて、ＡｌｅｘＮｅｔの暗号化及び復号化のオーバーヘッドは、暗号ハードウェア及びイントラ－レイヤパイプラインを使用して、大幅に減った他のＤＮＮよりも大きい。下記の表３は、ＧｕａｒｄｉａＮＮがＤＮＮの実行速度に与える影響を示す。

【００９２】

【表３】

Name	Pager	+Direct Conv	+SRAM Mgmt	+Crypt HW	+IntraLayer Pipelining
AlexNet [35]	1.00x	1.28x	2.81x	8.26x	9.92x
DeepID [69]	1.00x	1.12x	7.93x	13.25x	13.80x
FgptAuth [66]	1.00x	2.98x	7.89x	15.54x	16.76x
GAZEL [58]	1.00x	8.68x	20.59x	26.21x	26.54
MobileNetV1 [25]	1.00x	1.59x	3.31x	5.90x	6.54x
Smart Doll [11]	1.00x	8.44x	22.20x	30.58x	31.45x
Tiny Darknet [63]	1.00x	1.10x	4.11x	8.32x	9.35x
VGG-7 [37]	1.00x	3.59x	19.28x	25.26x	25.94x

【００９３】

上記表３に示すように、本発明で提案したＧｕａｒｄｉａＮＮはセキュリティ保証を損傷させることなく、広範囲なＤＮＮの実行を加速化することができる。

【００９４】

＜高エネルギーの効率（Ｈｉｇｈ　Ｅｎｅｒｇｙ　Ｅｆｆｉｃｉｅｎｃｙ）＞

【００９５】

ここで、ベンチマークで各ＤＮＮに対する全ての構成でＤＮＮ実行のエネルギー消費を調査する。先ず、遊休状態及びＤＮＮの実行中に装置全体の平均電力を測定し、ＤＮＮの実行による平均電力の増加を計算するために両値を引く。その次に、平均電力の増加と待ち時間とを掛けてＤＮＮ実行のエネルギー消費を計算する。正規化された結果は、図１５の（ｂ）に出ている。提案された各技法を適用することによってエネルギー消費が減少することを観察することができる。基本フレームワークと比較して、ＧｕａｒｄｉａＮＮのエネルギー消費は、幾何平均９２．３％減少し、エネルギー効率性が１５．２倍向上した。これは、ＧｕａｒｄｉａＮＮが提供するかなりの待ち時間の減少のためである。提案された技術を適用した後にも、ＤＮＮの実行中に装置の全体電力は同じレベルに維持される。減少した待ち時間と結合されたＧｕａｒｄｉａＮＮは、基本フレームワークよりもはるかに高いエネルギー効率性を達成する。

【００９６】

＜敏感度研究（Ｓｅｎｓｉｔｉｖｉｔｙ　Ｓｔｕｄｉｅｓ）＞

【００９７】

イントラ－レイヤパイプライニングのバルクサイズがＤＮＮの実行速度に与える影響を研究するために、４個のバルクサイズ（４、８、１６、及び３２の出力チャンネル）でＧｕａｒｄｉａＮＮのＤＮＮの実行待ち時間を測定する。図１６は、バルクサイズ４に正規化された測定値を示す。４個のバルクサイズ以外にも、各ＤＮＮを最低の待ち時間に導く最適なレイヤ別バルクサイズの集合を計算して、次のように待ち時間を測定した。イントラ－レイヤパイプライニングが最適なバルクサイズで待ち時間を最大１４．２４％向上させることを観察することができる（図１６の５番目の棒）。待ち時間が最も高いバルクサイズと比較する。殆どのＤＮＮの場合、バルクサイズが増加するにつれて、待ち時間が減少する傾向がある。バルクサイズが大きいほど暗号ハードウェアに対する暗号化及び復号化要求（及び割り込み）数が減少するためである。これは、パイプライン演算ステップの待ち時間を効果的に減らして、全体待ち時間を減らす。しかし、バルクサイズはＳＲＡＭ容量に拘束される。例えば、ＦｇｐｔＡｕｔｈに８よりも大きいバルクサイズを使用すると、必要なメモリサイズがＳＲＡＭサイズを超える。これが、バルクサイズが１６及び３２であるＦｇｐｔＡｕｔｈの実行待ち時間が図１６で空いている理由である。

【００９８】

ＧｕａｒｄｉａＮＮで使用するＤＭＡ可能暗号ハードウェアは、複数のブロック暗号及び作業モードをサポートする。ＧｕａｒｄｉａＮＮはセキュリティ強化のためにＡＥＳを使用し、ここではＡＥＳ－ＥＣＢ及びＡＥＳ－ＣＢＣの２つの作動モードを比較する。（

ａ）暗号ハードウェア、及び（ｂ）ＯＰ－ＴＥＥの基本ＣＰＵベースの暗号化ライブラリであるＬｉｂＴｏｍＣｒｙｐｔのあるＣＰＵで実行し、多様なキーサイズでＡＥＳ－ＥＣＢ及びＡＥＳ－ＣＢＣの暗号化及び復号化のスループットを測定する。図１７は測定されたスループットを示す。ＬｉｂＴｏｍＣｒｙｐｔを使用するＣＰＵでよりも、暗号ハードウェアでＡＥＳ－ＥＣＢ及びＡＥＳ－ＣＢＣがはるかに速く実行されることが観察された。また、キーサイズが長いほどＡＥＳの暗号化及び復号化でより多くのラウンドが発生するため、ＬｉｂＴｏｍ－Ｃｒｙｐｔを使用するＣＰＵのスループットは、キーサイズが増加するにつれて減少する。動作モードを比較すると、ＡＥＳ－ＣＢＣはチェーンによってＡＥＳ－ＥＣＢよりもスループットが少し少ない。逆に、暗号ハードウェアでは、キーサイズや作動モードがスループットに顕著な影響を与えない。これは、暗号化及び復号化性能が暗号ハードウェア及びＤＭＡバッファ管理コストによって制限されるためである。それでも、ＧｕａｒｄｉａＮＮで使用される暗号ハードウェアは、全てのケースでＣＰＵよりもはるかに高い暗号化及び復号化の効率性を提供する。

【００９９】

モバイル及び組み込み機器でＤＮＮの実行に対する関心が高まるにつれて、当該機器でＤＮＮの実行を加速化する様々な技術が登場している。しかし、連合学習のような装置内で処理されるデータは差分プライバシーに基づいているため、依然として多くのプライバシーの問題がある。従って、ＤＮＮにトラスト実行環境を活用することが合理的な方向である。例えば、ＳｅｃｕｒｅＴＦは、トラスト実行環境を活用するＴｅｎｓｏｒＦｌｏｗベースの分散セキュリティ機械学習フレームワークである。ＰＰＦＬは、ローカル教育及び集計、多重パートＭＬにトラスト実行環境を活用して、安全な連合学習を加速化する。Ｃｈｉｒｏｎ及びＭｙｅｌｉｎは、機械学習内でトラスト実行環境をサービスとして活性化する。

【０１００】

ＤａｒｋｎｅＴＺ、Ｉｎｆｅｎｃｌａｖｅ、及びＳｌａｌｏｍは、いずれもトラスト実行環境内でＤＮＮの一部を実行するように提案する。しかし、残りのレイヤを攻撃にさらして素直にアクセス方式を適用すると、かなりの量の性能オーバーヘッドが発生する。このような問題を解決するために、ＨｙｂｒｉｄＴＥＥはリモートサーバーのトラスト実行環境にＤＮＮの実行を要求することを提案する。しかし、ＨｙｂｒｉｄＴＥＥの加速効果は、ローカルトラスト実行環境のＤＮＮの実行を最適化していないため、あまり重要ではない。

【０１０１】

本発明は、ローカルトラスト実行環境でＤＮＮの完全な保護を提案するだけでなく、実際のモバイル及び組み込み環境で実行できるようにする驚くべき速度の向上を提供することができる。

【０１０２】

＜ＤＮＮ暗号化（ＤＮＮ　Ｅｎｃｒｙｐｔｉｏｎ）＞

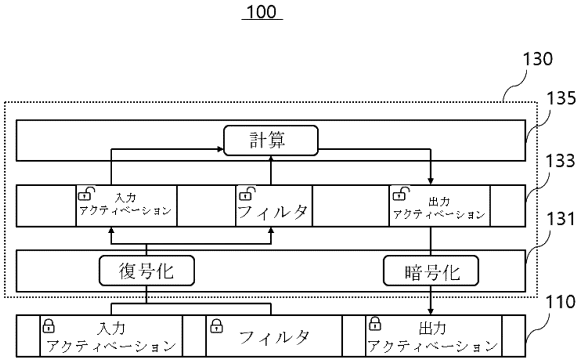
【０１０３】

ＤＮＮを保護するまた別の方向は、ＤＮＮデータを暗号化することである。例えば、ＳｅｃｕｒｅＭＬは、拡張可能な個人情報保護のＤＮＮフレームワークを構築するために、安全なマルチパーティ計算を活用した。ＳｏｆｔＭＥは信頼できる環境を提供し、暗号化と復号化及び演算ステップとで構成された信頼できる作業を実行する。ＳｏｆｔＭＥでＤＮＮを実行すると機密性が保証されるが、データの暗号化及び復号化にＣＰＵを使用して、大きな性能オーバーヘッドが発生する。その中で、準同型暗号を適用すると、暗号化されたデータに対する演算が可能であるため有望である。ＣｒｙｐｔｏＮｅｔｓは、そのようなアイデアが実現可能であることを示し、アイデアをセキュリティ教育に拡張する。ＭｉｎｉＯＮＮは、事前訓練されたＤＮＮを認識できないように変換する技術を提案する。また、準同型暗号化を活用して安全な連合遷移学習プロトコルを構築する。しかし、準同型暗号は演算スループットが低く、モバイル及び組み込み装置には実用的ではないと考えられることが多い。

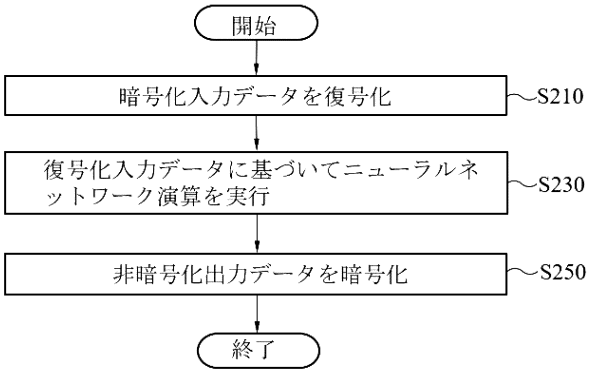
【０１０４】
ＴＥＥは、ＡＲＭ ＴｒｕｓｔＺｏｎｅ及びＩｎｔｅｌ ＳＧＸが広く使用される商用実装であって、高いセキュリティ保証のために注目を集めている。ソフトウェアベースのセキュリティソリューションが適用されるが、様々なアプリケーションを保護するために成功的に悪用された。しかし、キャッシュアーキテクチャ、二重インスタンスアプリ、又は重畳アプリのようなＴＥＥシステムを対象とする多くの脅威がある。これにより、セキュリティ強化のために、最近いくつかの提案がある。また、ＴＥＥの活用の困難を緩和するための多くの作業が提案されている。最小カーネルは、ＴＥＥの制限されたメモリ問題を解決するために小さなカーネルを構築する。ＣｏＳＭＩＸは、アプリケーションレベルのセキュリティページエラー処理器を許容する。ＴＥＥＭｏｎはＴＥＥのための性能モニタリングフレームワークである。
【０１０５】
本発明によるトラスト環境ベースの人工知能装置は、トラスト実行環境で人工ニューラルネットワークの実行を隔離して、動作速度の遅いＤＲＡＭに格納されたデータを暗号化してセキュリティを強化することができる。また、直接畳み込み及びＳＲＡＭ管理を通じてＤＲＡＭのアクセス数を減らすことができ、暗号ハードウェアでデータの暗号化及び復号化をオフロードし、パイプライニングを実現してニューラルネットワーク演算の実行をデータの暗号化及び復号化演算にオーバーラップするように実行して人工ニューラルネットワークの実行を加速化することができる。
【０１０６】
以上、本発明の実施形態について図面を参照しながら詳細に説明したが、本発明は、上述の実施形態に限定されるものではなく、本発明の技術思想から逸脱しない範囲内で多様に変更実施することが可能である。
【符号の説明】
【０１０７】
１００　（トラスト環境ベースの）人工知能装置
１１０　第１類型のメモリ
１３０　トラスト人工知能処理部
１３１　暗号処理フロント－エンドプロセス
１３３　第２類型のメモリ
１３５　プロセッサ

【図面】

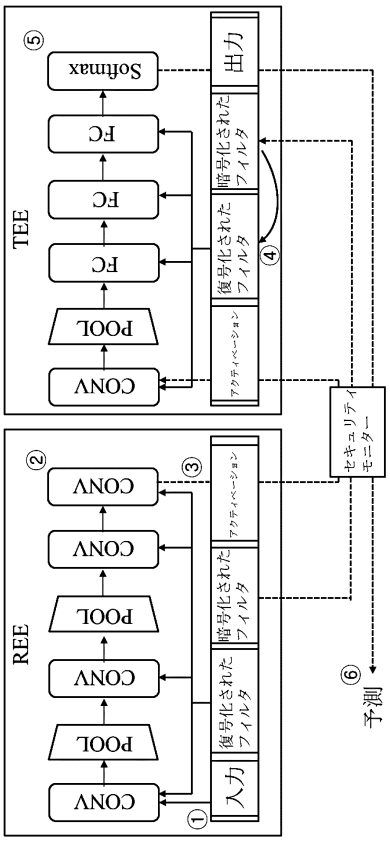
【図１】



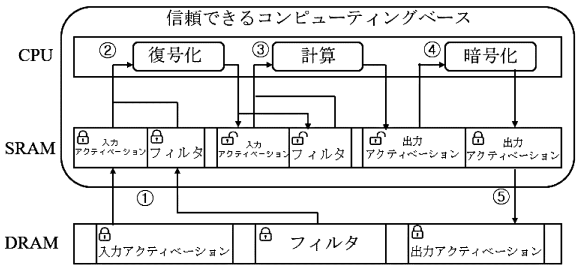
【図２】



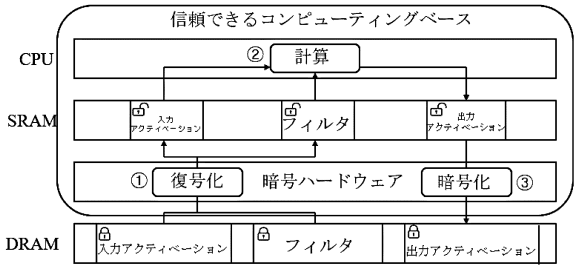
【図３】



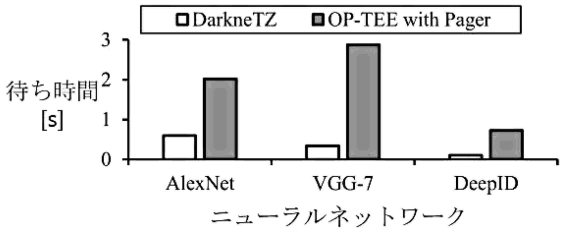
【図４ａ】



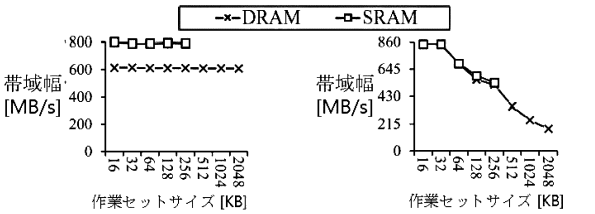
【図４ｂ】



【図５】

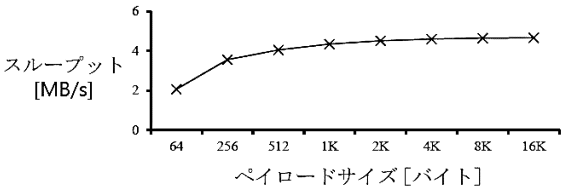


【図６】

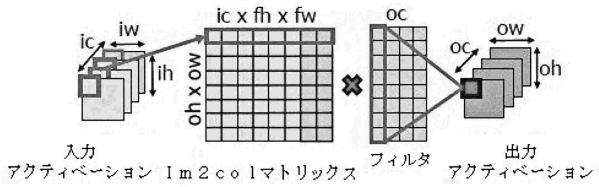


（ａ）シーケンシャル４－バイトアクセス　（ｂ）ランダム４－バイトアクセス

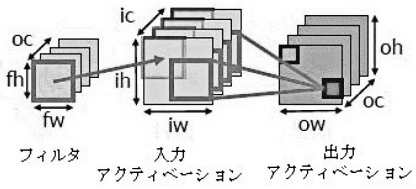
【図７】



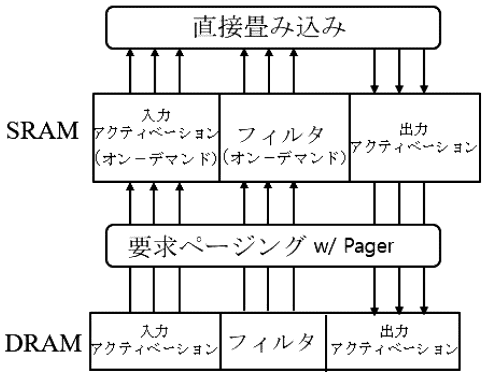
【図８ａ】



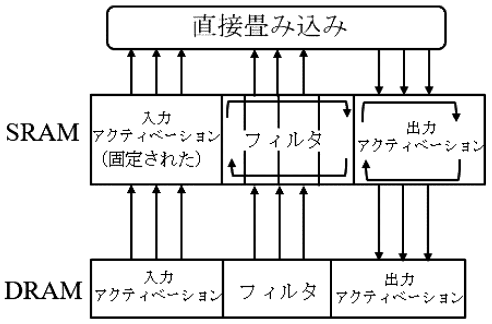
【図８ｂ】



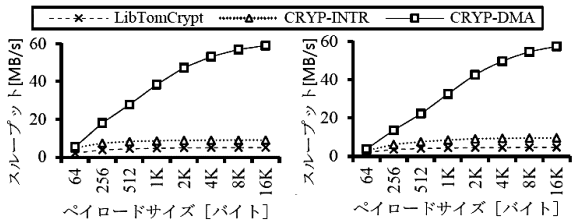
【図９ａ】



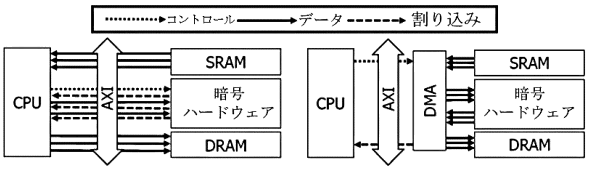
【図９ｂ】



【図１０】

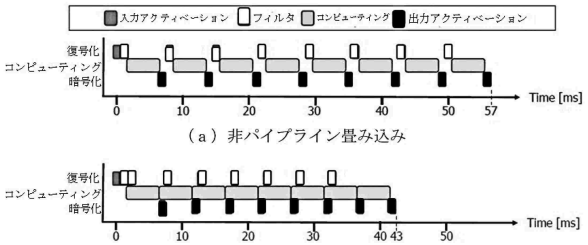


【図１１】



（ａ）割り込み駆動のオフロード　（ｂ）DMA駆動のオフロード

【図１２】



（ａ）非パイプライン畳み込み

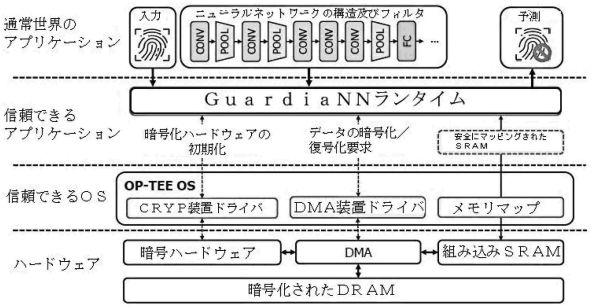
（ｂ）イントラーレイヤパイプライニングを使用したパイプライン畳み込み

【図１３】

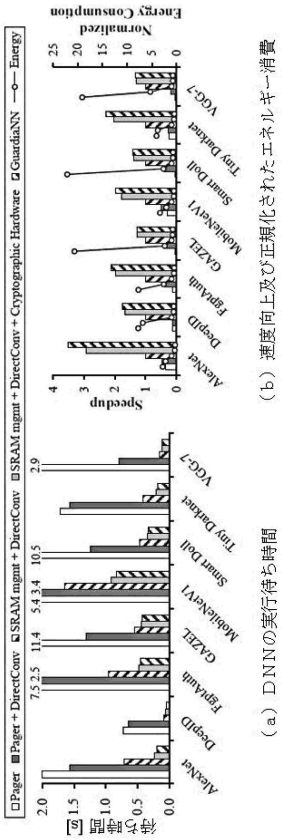
アルゴリズム 1イントラーレイヤパイプライニングのための畳み込みレイヤ

1: D_{OC}: The first output channel for each intra-layer pipelining stage
2: $CrypBlkSize$: Cryptographic hardware invocation granularity
3: $D_{filters,oact}$: DRAM addresses for filters and output activations; CHW format
4: S_{iact}: SRAM address for input activations; CHW format
5: $S_{filters,oact}$: In-SRAM circular queues for filters and output activations
6: f, o: # filters / # output activations of a bulk of output channels
7: F, O: # filters / # output activations to be requested to cryptographic hardware
8: $crypCtx$: the cryptographic context for the convolutional layer
9:
10: **procedure** PIPELINEDCONVLAYER
11: $F = \lceil f/CrypBlkSize \rceil * CrypBlkSize$
12: $DecryptData(crypCtx, D_{filter}, S_{filter}.tail, F)$
13: $S_{filter}.tail += F$
14: **for** oc in D_{OC} **do**
15: **if** $(S_{filter}.tail - S_{filter}.head) == 0$ **then**
16: $F = \lceil f/CrypBlkSize \rceil * CrypBlkSize$
17: **else**
18: $leftover = (S_{filter}.tail - S_{filter}.head - f)$
19: $F = \lceil (f + leftover)/CrypBlkSize \rceil * CrypBlkSize$
20: **end if**
21: $DecryptData(crypCtx, D_{filter}[oc], S_{filter}.tail, F)$
22: $S_{filter}.tail = S_{filter}.tail + F$
23: $DirectConv(S_{iact}, S_{filter}.head, S_{oact}.tail)$
24: $S_{filter}.head = S_{filter}.head + f$
25: $S_{oact}.tail = S_{oact}.tail + o$
26: $O = \lfloor (S_{oact}.tail - S_{oact}.head)/CrypBlkSize \rfloor * CrypBlkSize$
27: $EncryptData(crypCtx, S_{oact}.head, D_{oact}[oc], O)$
28: $S_{oact}.head = S_{oact}.head + O$
29: **end for**
30: **end procedure**

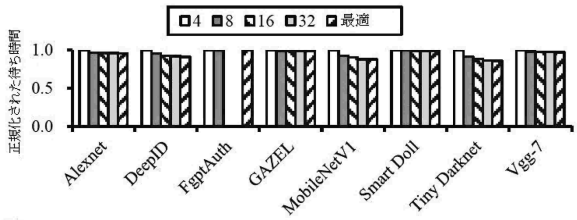
【図１４】



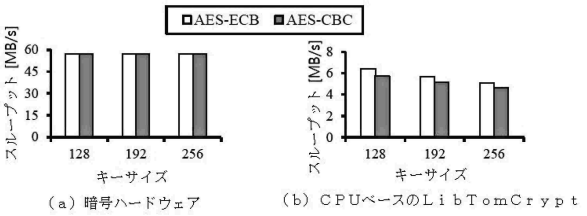
【図１５】



【図１６】



【図１７】



フロントページの続き

５０
(72)発明者　チェ， ジン ウ
大韓民国 ソウル特別市 西大門区 延世路 ５０
(72)発明者　イム， チェ ミン
大韓民国 ソウル特別市 西大門区 延世路 ５０
(72)発明者　イ， ス ヒョン
大韓民国 ソウル特別市 西大門区 延世路 ５０
(72)発明者　ソン， ド キョン
大韓民国 ソウル特別市 西大門区 延世路 ５０
(72)発明者　イ， ジン ホ
大韓民国 ソウル特別市 麻浦区 ワールド カップ ブクーロ ２３５

審査官　青木 重徳

(56)参考文献
米国特許第１０９５６５８４（ＵＳ，Ｂ１）
特開２００５－０１８４３４（ＪＰ，Ａ）
中井 綱人，"ＯＰ－ＴＥＥを用いた隔離ＡＩハードウェアアクセラレーションの実装評価"，２０２２年 暗号と情報セキュリティシンポジウム予稿集，2022年01月21日，pp.1-8
Yunjie Deng et al.，StrangBox: A GPU TEE on Arm Endpoints，CCS'22: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security，2022年11月07日，pp. 769-783
Meiyu Zhang et al.，SoftME: A Software-Based Memory Protection Approach for TEE System to Resist Physical Attacks，Security and Communication Networks，Hindawi [オンライン]，2019年01月，Volume 2019, Issue 1，(2024.10.29 検索)、インターネット，URL: https://onlinelibrary.wiley.com/doi/epdf/10.1155/2019/8690853

(58)調査した分野　(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　２１／６０
Ｇ０６Ｆ　２１／６２
Ｇ０６Ｎ　３／０２
ＪＳＴＰｌｕｓ／ＪＭＥＤＰｌｕｓ／ＪＳＴ７５８０（ＪＤｒｅａｍＩＩＩ）
ＩＥＥＥ Ｘｐｌｏｒｅ
ＴＨＥ ＡＣＭ ＤＩＧＩＴＡＬ ＬＩＢＲＡＲＹ