(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6829327号
(P6829327)
(45)発行日　令和3年2月10日(2021.2.10)　(24)登録日　令和3年1月25日(2021.1.25)

(51)Int.Cl.　F I
G06N 3/10 (2006.01)　G06N 3/10

請求項の数 13　(全 26 頁)

(21)出願番号　特願2019-564530(P2019-564530)
(86)(22)出願日　平成31年3月29日(2019.3.29)
(65)公表番号　特表2020-532778(P2020-532778A)
(43)公表日　令和2年11月12日(2020.11.12)
(86)国際出願番号　PCT/CN2019/080510
(87)国際公開番号　WO2020/029592
(87)国際公開日　令和2年2月13日(2020.2.13)
審査請求日　令和1年11月18日(2019.11.18)
(31)優先権主張番号　201810913895.6
(32)優先日　平成30年8月10日(2018.8.10)
(33)優先権主張国・地域又は機関
中国(CN)

(73)特許権者　519411401
カンブリコン　テクノロジーズ　コーポレイション　リミティド
中華人民共和国，ベイジン　１００１９０，ハイジャン　ディストリクト，コーシュエユアン　サウス　ロード，ナンバー　６，サイエンティフィック　リサーチ　ビルディング，スイート　６４４
(74)代理人　100099759
弁理士　青木　篤
(74)代理人　100123582
弁理士　三橋　真二
(74)代理人　100114018
弁理士　南山　知広

最終頁に続く

(54)【発明の名称】変換方法、装置、コンピューターデバイス及び記憶媒体

(57)【特許請求の範囲】
【請求項１】
モデル変換方法であって、
初期オフラインモデルとコンピューターデバイスのハードウェア属性情報を獲得するステップと、
前記初期オフラインモデルと前記コンピューターデバイスのハードウェア属性情報に従って、前記初期オフラインモデルのモデル属性情報と前記コンピューターデバイスのハードウェア属性情報が一致するかどうかを判断するステップと、及び
前記初期オフラインモデルのモデル属性情報と前記コンピューターデバイスのハードウェア属性情報が一致しない場合、前記コンピューターデバイスのハードウェア属性情報と予め設定されたモデル変換ルールに従って、前記初期オフラインモデルを前記コンピューターデバイスのハードウェア属性情報と一致するターゲットオフラインモデルに変換するステップとを含む、前記モデル変換方法。
【請求項２】
前記コンピューターデバイスのハードウェア属性情報と予め設定されたモデル変換ルールに従って、前記初期オフラインモデルを前記コンピューターデバイスのハードウェア属性情報と一致するターゲットオフラインモデルに変換するステップは、
前記コンピューターデバイスのハードウェア属性情報に従って、前記ターゲットオフラインモデルのモデル属性情報を決定するステップと、
前記初期オフラインモデルのモデル属性情報と前記ターゲットオフラインモデルのモデ

ル属性情報に従って、複数の前記予め設定されたモデル変換ルールから一つの前記モデル変換ルールをターゲットモデル変換ルールとして選択するステップと、及び
前記初期オフラインモデルのモデル属性情報と前記ターゲットモデル変換ルールに従って、前記初期オフラインモデルを前記ターゲットオフラインモデルに変換するステップとを含むことを特徴とする
請求項１に記載のモデル変換方法。
【請求項３】
前記初期オフラインモデルのモデル属性情報と前記ターゲットオフラインモデルのモデル属性情報に従って、複数の前記予め設定されたモデル変換ルールから一つの前記モデル変換ルールをターゲットモデル変換ルールとして選択するステップは、
前記初期オフラインモデルのモデル属性情報と前記ターゲットオフラインモデルのモデル属性情報に従って、複数の前記予め設定されたモデル変換ルールから一つ以上の使用可能なモデル変換ルールを選択するステップと、及び
前記一つ以上の使用可能なモデル変換ルールに優先順位を付け、優先順位の高い使用可能なモデル変換ルールを前記ターゲット変換ルールとして使用するステップとを含むことを特徴とする
請求項２に記載のモデル変換方法。
【請求項４】
前記一つ以上の使用可能なモデル変換ルールに優先順位を付けるステップは、
各前記使用可能なモデル変換ルールを使用して、前記初期オフラインモデルを前記ターゲットオフラインモデルに変換するためのプロセスパラメーターをそれぞれ獲得し、ここで、前記プロセスパラメーターは、変換速度、消費電力、メモリ使用率及びディスクＩ／Ｏ占有率中の一つまたは複数を含むステップと、及び
各前記使用可能なモデル変換ルールのプロセスパラメーターに従って、一つ以上の前記使用可能なモデル変換ルールに優先順位を付けるステップとを含むことを特徴とする
請求項３に記載のモデル変換方法。
【請求項５】
前記初期オフラインモデルのモデル属性情報、前記ターゲットオフラインモデルのモデル属性情報及び前記使用可能なモデル変換ルールの三つの間は１対１に対応するように保存されることを特徴とする
請求項３に記載のモデル変換方法。
【請求項６】
前記初期オフラインモデルと前記コンピューターデバイスのハードウェア属性情報に従って、前記初期オフラインモデルのモデル属性情報と前記コンピューターデバイスのハードウェア属性情報が一致するかどうかを判断するステップは、
前記初期オフラインモデルのモデル属性情報と前記コンピューターデバイスのハードウェア属性情報に従って、前記コンピューターデバイスが前記初期オフラインモデルの実行をサポートできること決定する場合、前記初期オフラインモデルのモデル属性情報と前記コンピューターデバイスのハードウェア属性情報が一致すると判定するステップと、
前記初期オフラインモデルのモデル属性情報と前記コンピューターデバイスのハードウェア属性情報に従って、前記コンピューターデバイスが前記初期オフラインモデルの実行をサポートしないと決定する場合、前記初期オフラインモデルのモデル属性情報と前記コンピューターデバイスのハードウェア属性情報が一致しないと判定するステップとを含むことを特徴とする
請求項１～５のいずれか一項に記載のモデル変換方法。
【請求項７】
前記初期オフラインモデルを獲得するステップは、
前記コンピューターデバイス上のアプリケーションソフトウェアを介して前記初期オフラインモデルと前記コンピューターデバイスのハードウェア属性情報を獲得するステップを含むことを特徴とする

請求項１～５のいずれか一項に記載のモデル変換方法。
【請求項８】
前記ターゲットオフラインモデルを前記コンピューターデバイスの第１メモリまたは第２メモリに保存するステップをさらに含むことを特徴とする
請求項１～５のいずれか一項に記載のモデル変換方法。
【請求項９】
前記ターゲットオフラインモデルを獲得し、前記ターゲットオフラインモデルを実行し、ここで、前記ターゲットオフラインモデルは、オリジナルネットワークにおける各オフラインノードに対応するネットワークの重み、命令及び各オフラインノードと前記オリジナルネットワーク中の他の計算ノードとの間のインターフェースデータを含むステップをさらに含むことを特徴とする
請求項８に記載のモデル変換方法。
【請求項１０】
モデル変換装置であって、
初期オフラインモデルとコンピューターデバイスのハードウェア属性情報を獲得するために使用される獲得モジュールと、
前記初期オフラインモデルと前記コンピューターデバイスのハードウェア属性情報に従って、前記初期オフラインモデルのモデル属性情報と前記コンピューターデバイスのハードウェア属性情報が一致するかどうかを判断するように使用される判断モジュールと、及び
前記初期オフラインモデルのモデル属性情報と前記コンピューターデバイスのハードウェア属性情報が一致しない場合、前記コンピューターデバイスのハードウェア属性情報と予め設定されたモデル変換ルールに従って、前記初期オフラインモデルを前記コンピューターデバイスのハードウェア属性情報と一致するターゲットオフラインモデルに変換するように使用される変換モジュールとを含む、前記モデル変換装置。
【請求項１１】
コンピュータプログラムが保存されるメモリとプロセッサを含むコンピューターデバイスであって、前記プロセッサが前記コンピュータプログラムを実行する場合、請求項１～７のいずれか一項に記載の方法を実現する、前記コンピューターデバイス。
【請求項１２】
前記プロセッサは、演算ユニットとコントローラーユニットを含み、前記演算ユニットは、メイン処理回路と複数のスレーブ処理回路を含み、
前記コントローラーユニットは、データ、機械学習モデル及び計算命令を獲得するために使用され、
前記コントローラーユニットは、さらに前記計算命令を解析して複数の演算命令を獲得し、前記複数の演算命令及び前記データを前記メイン処理回路に送信するように使用され、
前記メイン処理回路は、前記データ及び前記メイン処理回路と前記複数のスレーブ処理回路との間で伝送されるデータと演算命令に対して前処理を実行するように使用され、
前記複数のスレーブ処理回路は、前記メイン処理回路から伝送されたデータ及び演算命令に従って、中間演算を実行して複数の中間結果を獲得し、複数の中間結果を前記メイン処理回路に伝送するように使用され、
前記メイン処理回路は、さらに前記複数の中間結果に対して後続処理を実行して、前記計算命令の計算結果を獲得するように使用されることを特徴とする
請求項１１に記載のコンピューターデバイス。
【請求項１３】
コンピュータプログラムが保存されたコンピュータ可読記憶媒体であって、前記コンピュータプログラムは、プロセッサによって実行される場合、請求項１～７のいずれか一項に記載の方法のステップを実行する、前記コンピュータ可読記憶媒体。
【発明の詳細な説明】

【技術分野】
【０００１】
関連出願
本願発明は、出願日が２０１８年０８月１０日であり、出願番号が２０１８１０９１３８９５．６であり、名前が「変換方法、装置、コンピューターデバイス及び記憶媒体」である中国特許出願に対して優先権を主張し、当該中国特許出願の全ての内容は参照により本明細書に組み込まれる。
【０００２】
本発明は、コンピュータ技術分野に関し、具体的に、変換方法、装置、コンピューターデバイス及び記憶媒体に関する。
【背景技術】
【０００３】
人工知能技術の開発により、ディープラーニングはユビキタスで不可欠になり、例えば、ＴｅｎｓｏｒＦｌｏｗ、ＭＸＮｅｔ、Ｃａｆｆｅ及びＰｙＴｏｒｃｈなどの多くのスケーラブルなディープラーニングシステムが生成され、上記ディープラーニングシステムは、ＣＰＵやＧＰＵなどのプロセッサで実行することができるニューラルネットワークモデルまたは他の機械学習モデルを提供するために使用されることができる。ニューラルネットワークモデルなどの機械学習モデルが異なるプロセッサで実行される必要がある場合、多くの場合、ニューラルネットワークモデルなどの機械学習モデルを変換する必要がある。
【０００４】
従来の変換方法は次の通りである。開発者は、同じニューラルネットワークモデルに複数の異なる変換モデルを配置することができ、複数の変換モデルは、異なるプロセッサに適用されることができ、実際の使用において、コンピューターデバイスは、当該ニューラルネットワークモデル及びそれに対応する複数の変換モデルを同時に受信する必要があり、ユーザーは現在のコンピューターデバイスのタイプに従って、上記の複数のモデルから一つのモデルを選択して、当該ニューラルネットワークモデルが現在のコンピューターデバイスで実行できるようにする必要がある。しかし、上記変換方法のデータ入力量とデータ処理量はともに大きく、上記比較的に大きいデータ入力量及びデータ処理量は、コンピューターデバイスの記憶容量及び処理制限を容易に超えてしまい、コンピューターデバイスの処理速度が低下し、さらに正常に動作できないことがある。
【発明の概要】
【発明が解決しようとする課題】
【０００５】
関連技術に存在する問題を少なくともある程度克服するために、本出願は、変換方法、装置、コンピューターデバイス及び記憶媒体を提供する。
【課題を解決するための手段】
【０００６】
本出願は、初期オフラインモデルとコンピューターデバイスのハードウェア属性情報を獲得するステップと、前記初期オフラインモデルと前記コンピューターデバイスのハードウェア属性情報に従って、前記初期オフラインモデルのモデル属性情報と前記コンピューターデバイスのハードウェア属性情報が一致するかどうかを判断するステップと、及び前記初期オフラインモデルのモデル属性情報と前記コンピューターデバイスのハードウェア属性情報が一致しない場合、前記コンピューターデバイスのハードウェア属性情報と予め設定されたモデル変換ルールに従って、前記初期オフラインモデルを前記コンピューターデバイスのハードウェア属性情報と一致するターゲットオフラインモデルに変換するステップとを含むモデル変換方法を提供する。
【０００７】
一つの実施例において、前記コンピューターデバイスのハードウェア属性情報と予め設定されたモデル変換ルールに従って、前記初期オフラインモデルを前記コンピューターデ

バイスのハードウェア属性情報と一致するターゲットオフラインモデルに変換するステップは、前記コンピューターデバイスのハードウェア属性情報に従って、前記ターゲットオフラインモデルのモデル属性情報を決定するステップと、前記初期オフラインモデルのモデル属性情報と前記ターゲットオフラインモデルのモデル属性情報に従って、複数の前記予め設定されたモデル変換ルールから一つの前記モデル変換ルールをターゲットモデル変換ルールとして選択するステップと、及び前記初期オフラインモデルのモデル属性情報と前記ターゲットモデル変換ルールに従って、前記初期オフラインモデルを前記ターゲットオフラインモデルに変換するステップとを含む。

【０００８】

一つの実施例において、前記初期オフラインモデルのモデル属性情報と前記ターゲットオフラインモデルのモデル属性情報に従って、複数の前記予め設定されたモデル変換ルールから一つの前記モデル変換ルールをターゲットモデル変換ルールとして選択するステップは、前記初期オフラインモデルのモデル属性情報と前記ターゲットオフラインモデルのモデル属性情報に従って、複数の前記予め設定されたモデル変換ルールから一つ以上の使用可能なモデル変換ルールを選択するステップと、及び前記一つ以上の使用可能なモデル変換ルールに優先順位を付け、優先順位の高い使用可能なモデル変換ルールを前記ターゲット変換ルールとして使用するステップとを含む。

【０００９】

一つの実施例において、前記一つ以上の使用可能なモデル変換ルールに優先順位を付けるステップは、各前記使用可能なモデル変換ルールを使用して、前記初期オフラインモデルを前記ターゲットオフラインモデルに変換するためのプロセスパラメーターをそれぞれ獲得し、ここで、前記プロセスパラメーターは、変換速度、消費電力、メモリ使用率及びディスクＩ／Ｏ占有率中の一つまたは複数を含むステップと、及び各前記使用可能なモデル変換ルールのプロセスパラメーターに従って、一つ以上の前記使用可能なモデル変換ルールに優先順位を付けるステップとを含む。

【００１０】

一つの実施例において、前記初期オフラインモデルのモデル属性情報、前記ターゲットオフラインモデルのモデル属性情報及び前記使用可能なモデル変換ルールの三つの間は１対１に対応するように保存される。

【００１１】

一つの実施例において、前記初期オフラインモデルと前記コンピューターデバイスのハードウェア属性情報に従って、前記初期オフラインモデルのモデル属性情報と前記コンピューターデバイスのハードウェア属性情報が一致するかどうかを判断するステップは、前記初期オフラインモデルのモデル属性情報と前記コンピューターデバイスのハードウェア属性情報に従って、前記コンピューターデバイスが前記初期オフラインモデルの実行をサポートできること決定する場合、前記初期オフラインモデルのモデル属性情報と前記コンピューターデバイスのハードウェア属性情報が一致すると判定するステップと、前記初期オフラインモデルのモデル属性情報と前記コンピューターデバイスのハードウェア属性情報に従って、前記コンピューターデバイスが前記初期オフラインモデルの実行をサポートしないと決定する場合、前記初期オフラインモデルのモデル属性情報と前記コンピューターデバイスのハードウェア属性情報が一致しないと判定するステップとを含む。

【００１２】

一つの実施例において、前記初期オフラインモデルを獲得するステップは、前記コンピューターデバイス上のアプリケーションソフトウェアを介して前記初期オフラインモデルと前記コンピューターデバイスのハードウェア属性情報を獲得するステップを含む。

【００１３】

一つの実施例において、前記ターゲットオフラインモデルを前記コンピューターデバイスの第１メモリまたは第２メモリに保存するステップをさらに含む。

【００１４】

一つの実施例において、前記方法は、前記ターゲットオフラインモデルを獲得し、前記

ターゲットオフラインモデルを実行し、ここで、前記ターゲットオフラインモデルは、オリジナルネットワークにおける各オフラインノードに対応するネットワークの重み、命令及び各オフラインノードと前記オリジナルネットワーク中の他の計算ノードとの間のインターフェースデータを含むステップをさらに含む。

【００１５】

本出願は、初期オフラインモデルとコンピューターデバイスのハードウェア属性情報を獲得するために使用される獲得モジュールと、前記初期オフラインモデルと前記コンピューターデバイスのハードウェア属性情報に従って、前記初期オフラインモデルのモデル属性情報と前記コンピューターデバイスのハードウェア属性情報が一致するかどうかを判断するように使用される判断モジュールと、及び前記初期オフラインモデルのモデル属性情報と前記コンピューターデバイスのハードウェア属性情報が一致しない場合、前記コンピューターデバイスのハードウェア属性情報と予め設定されたモデル変換ルールに従って、前記初期オフラインモデルを前記コンピューターデバイスのハードウェア属性情報と一致するターゲットオフラインモデルに変換するように使用される変換モジュールとを含むモデル変換装置を提供する。

【００１６】

本出願は、コンピュータプログラムが保存されるメモリとプロセッサを含み、前記プロセッサが前記コンピュータプログラムを実行する場合、上記の変換方法を実現するコンピューターデバイスを提供する。

【００１７】

一つの実施例において、前記プロセッサは、演算ユニットとコントローラーユニットを含み、前記演算ユニットは、メイン処理回路と複数のスレーブ処理回路を含み、前記コントローラーユニットは、データ、機械学習モデル及び計算命令を獲得するために使用され、前記コントローラーユニットは、さらに前記計算命令を解析して複数の演算命令を獲得し、前記複数の演算命令及び前記データを前記メイン処理回路に送信するように使用され、前記メイン処理回路は、前記データ及び前記メイン処理回路と前記複数のスレーブ処理回路との間で伝送されるデータと演算命令に対して前処理を実行するように使用され、前記複数のスレーブ処理回路は、前記メイン処理回路から伝送されたデータ及び演算命令に従って、中間演算を実行して複数の中間結果を獲得し、複数の中間結果を前記メイン処理回路に伝送するように使用され、前記メイン処理回路は、さらに前記複数の中間結果に対して後続処理を実行して、前記計算命令の計算結果を獲得するように使用される。

【００１８】

本出願は、コンピュータプログラムが保存され、前記コンピュータプログラムは、プロセッサによって実行される場合、上記の変換方法のステップを実行するコンピュータ可読記憶媒体を提供する。

【００１９】

以上の一般的な説明及び以下の詳細な説明は単なる例示及び解釈であり、本発明を限定する意図ではないことを理解されたい。

【図面の簡単な説明】

【００２０】

【図１】一つの実施例におけるコンピューターデバイスの構造ブロック図である。
【図２】図１におけるプロセッサの一つの実施例の構造ブロック図である。
【図３】図１におけるプロセッサの一つの実施例の構造ブロック図である。
【図４】図１におけるプロセッサの一つの実施例の構造ブロック図である。
【図５】一つの実施例におけるモデル変換方法のフロー模式図である。
【図６】一つの実施例におけるモデル変換方法のフロー模式図である。
【図７】一つの実施例におけるオフラインモデル生成方法のフロー模式図である。
【図８】一つの実施例におけるオフラインモデル生成方法のフロー模式図である。
【図９】一実施例のネットワークモデルのネットワーク構造図である。
【図１０】図９におけるネットワークモデルのオフラインモデル生成プロセス模式図であ

る。
【発明を実施するための形態】
【００２１】
本明細書の図面は、本明細書に組み込まれ、その一部を構成し、本出願の実施例を例示し、明細書とともに本出願の原理を説明するために使用される。
【００２２】
ここで、例示的な実施例を詳細に説明し、その例を添付の図面に示す。以下の説明が図面に関する場合、特に明記しない限り、異なる図面の同じ数字は、同じまたは類似の要素を示す。以下の例示的な実施例で説明される実施形態は、本出願と一致するすべての実施形態を表すもためはない。代わりに、それらは、添付の特許請求の範囲に詳述されている、本出願のいくつかの態様と一致する装置及び方法の単なる例である。
【００２３】
図１は、一実施例におけるコンピューターデバイスのブロック図であり、当該コンピューターデバイスは、携帯電話またはタブレットコンピュータなどのモバイル端末、またはデスクトップコンピュータ、タブレットカード、またはクラウドサーバなどの端末であり得る。当該コンピューターデバイスは、ロボット、プリンター、スキャナー、運転レコーダー、ナビゲーター、カメラ、ビデオカメラ、プロジェクター、時計、モバイルストレージ、ウェアラブルデバイス、乗り物、家電製品、および／または医療機器に適用できる。ここで、乗り物は、飛行機、船、および／または車両を含むことができ、家電は、テレビ、エアコン、電子レンジ、冷蔵庫、炊飯器、加湿器、洗濯機、電灯、ガスコンロ、レンジフードを含むことができ、医療機器は、核磁気共鳴装置、Ｂ超音波および／または心電計などを含むことができる。
【００２４】
当該コンピューターデバイスは、プロセッサ１００、当該プロセッサ１００に接続される第１メモリ２００及び第２メモリ３００を含むことができる。選択的に、プロセッサ１００は、ＣＰＵ（Ｃｅｎｔｒａｌ　Ｐｒｏｃｅｓｓｉｎｇ　Ｕｎｉｔ、中央プロセッサ）、ＧＰＵ（Ｇｒａｐｈｉｃｓ　Ｐｒｏｃｅｓｓｉｎｇ　Ｕｎｉｔ、グラフィックプロセッサ）またはＤＳＰ（Ｄｉｇｉｔａｌ　Ｓｉｇｎａｌ　Ｐｒｏｃｅｓｓｉｎｇ、デジタル信号処理）のような汎用プロセッサであることができ、当該プロセッサ１００は、ＩＰＵ（Ｉｎｔｅｌｌｉｇｅｎｃｅ　Ｐｒｏｃｅｓｓｉｎｇ　Ｕｎｉｔ、インテリジェントプロセッサ）などの専用ニューラルネットワークプロセッサであることもできる。もちろん、当該プロセッサは、命令セットプロセッサ、関連チップセット、専用マイクロプロセッサ（例えば、専用集積回路（ＡＳＩＣ））またはキャッシュ（ｃａｃｈｅ）用途に使用されるオンボードメモリなどであることもである。
【００２５】
選択的に、図２に示したように、当該プロセッサ１００は、コントローラーユニット１１０と演算ユニット１２０を含むことができ、ここで、コントローラーユニット１１０は、演算ユニット１２０に接続され、当該演算ユニット１２０は、一つのメイン処理回路１２１と複数のスレーブ処理回路１２２を含むことができる。コントローラーユニット１１０は、データ、機械学習モデル及び計算命令を獲得するために使用される。当該機械学習モデルは、具体的にネットワークモデルを含むことができ、当該ネットワークモデルは、ニューラルネットワークモデル及び／または非ニューラルネットワークモデルであることができる。コントローラーユニット１１０は、さらに獲得された計算命令を解析して演算命令を獲得し、複数の演算命令及びデータをメインプロセッサ回路に送信するために使用される。メイン処理回路は、データ及び当該メイン処理回路と複数のスレーブ処理回路との間で伝送されるデータ及び演算命令に対して前処理を実行するために使用される。複数のスレーブ処理回路は、メイン処理回路から伝送されたデータ及び演算命令に従って、中間演算を並列に実行して複数の中間結果を獲得し、複数の中間結果をメイン処理回路に伝送するために使用され、メイン処理回路は、さらに複数の中間結果に対して後続処理を実行して計算命令の計算結果を獲得するために使用される。

【００２６】
選択的に、当該コントローラーユニット１１０は、命令記憶ユニット１１１、命令処理ユニット１１２及び記憶キューユニット１１４を含むことができ、命令記憶ユニット１１１は、機械学習モデルに関連する計算命令を保存するために使用され、命令処理ユニット１１２は、計算命令に対して解析して複数の演算命令を獲得するために使用され、記憶キューユニット１１４は、命令キューを保存するために使用され、当該命令キューは、当該キューの前後順序によって実行される複数の演算命令または計算命令を含む。選択的に、当該コントローラーユニット１１０は、複数の演算命令を備える場合、第１演算命令と第１演算命令との間の第０演算命令に関連関係があるかどうかを決定するための依存関係処理ユニット１１３をさらに含み、例えば、第１演算命令と第０演算命令に関連関係が存在する場合、第１演算命令を命令記憶ユニット内にキャッシュし、第０演算命令が実行完了した後、命令記憶ユニットから第１演算命令を抽出して演算ユニットに伝送する。具体的に、依存関係処理ユニット１１３が第１演算命令に従って、第１演算命令における必要なデータ（例えば、マトリックス）の第１ストレージアドレス区間を抽出し、第０演算命令に従って、第０演算命令における必要なマトリックスの第０ストレージアドレス区間を抽出し、例えば、第１ストレージアドレス区間と第０ストレージアドレス区間に重なる領域がある場合、第１演算命令と第０演算命令に関連関係があると決定し、例えば、第１ストレージアドレス区間と第０ストレージアドレス区間に重なる領域がない場合、第１演算命令と第０演算命令に関連関係がないと決定する。

【００２７】
一つの実施例において、図３に示したように、演算ユニット１２０は、分岐処理回路１２３をさらに含むことができ、ここで、メイン処理回路１２１は分岐処理回路１２３に接続され、分岐処理回路１２３は複数のスレーブ処理回路１２２に接続され、分岐処理回路１２３は、メイン処理回路１２１とスレーブ処理回路１２２との間のデータまたは命令を転送するために使用される。この実施例において、メイン処理回路１２１は、具体的に一つの入力ニューロンを複数のデータブロックに割り当て、複数のデータブロックにおける少なくとも一つの少なくとも一つのデータブロック、重み及び複数の演算命令における少なくとも一つの演算命令を分岐処理回路に送信するために使用され、分岐処理回路１２３は、メイン処理回路１２１と複数のスレーブ処理回路１２２との間のデータブロック、重み及び演算命令を転送するために使用され、複数のスレーブ処理回路１２２は、当該演算命令に従って受信されたデータブロック及び重みに対して演算して中間結果を獲得し、中間結果を分岐処理回路１２３に伝送するために使用され、メイン処理回路１２１は、さらに分岐処理回路によって送信された中間結果を後続処理して当該計算命令の結果を獲得し、当該計算命令の結果を前記コントローラーユニットに送信するために使用される。

【００２８】
他の選択可能な実施例において、図４に示したように、演算ユニット１２０は、一つのメイン処理回路１２１と複数のスレーブ処理回路１２２を含むことができる。ここで、複数のスレーブ処理回路は、アレイ配置され、それぞれのスレーブ処理回路は隣接する他のスレーブ処理回路に接続され、メイン処理回路は、複数のスレーブ処理回路におけるｋ個のスレーブ処理回路に接続され、ｋ個のスレーブ処理回路は、第１行のｎ個のスレーブ処理回路、第ｍ行のｎ個のスレーブ処理回路及び第１列のｍ個のスレーブ処理回路であり、説明すべきのは、図１Ｃに示したＫ個のスレーブ処理回路は、第１行のｎ個のスレーブ処理回路、第ｍ行のｎ個のスレーブ処理回路及び第１列のｍ個のスレーブ処理回路のみを含み、即ち、当該ｋ個のスレーブ処理回路は、複数のスレーブ処理回路におけるメイン処理回路に直接に接続されるスレーブ処理回路である。Ｋ個のスレーブ処理回路は、メイン処理回路と複数のスレーブ処理回路との間でデータ及び命令を転送するために使用される。

【００２９】
選択的に、上記のメイン処理回路１２１は、変換処理回路、起動処理回路、加算処理回路中の一つまたは任意の組合せを含むことができ、変換処理回路は、メイン処理回路が受信したデータブロックまたは中間結果によって第１データ構造と第２データ構造との間の

交換（例えば、連続データと離散データの変換）を実行するために使用され、またはメイン処理回路が受信したデータブロックまたは中間結果によって第１データタイプと第２データタイプとの間の交換（例えば、固定小数点タイプと浮動小数点タイプの変換）を実行するために使用され、起動処理回路は、メイン処理回路内のデータの起動演算を実行するために使用され、加算処理回路は、加算演算または累積演算を実行するために使用される。

【００３０】

当該第１メモリ２００または第２メモリ３００は本出願の実施例で提供されるモデル変換方法を実行するためのコンピュータプログラムを保存することができる。具体的に、当該モデル変換方法は、コンピューターデバイスが受信した初期オフラインモデルのモデル属性情報と当該コンピューターデバイスのハードウェア属性情報が一致しない場合、当該コンピューターデバイスが当該初期オフラインモデルを実行することができるように、当該初期オフラインモデルを当該コンピューターデバイスのハードウェア属性情報と一致するターゲットオフラインモデルに変換する。上記のモデル変換方法において、コンピューターデバイスのデータ入力量が少なく、人間の介入なしで初期オフラインモデルからターゲットオフラインモデルへの変換を自動的に完了することができ、変換プロセスが簡単で、変換効率が高い。さらに、第１メモリ２００または第２メモリ３００に保存されたコンピュータプログラムは、さらに本出願の実施例で提供されるオフラインモデル生成方法を実現するために使用される。

【００３１】

選択的に、第１メモリ２００は、ネットワーク入力データ、ネットワーク出力データ、ネットワークの重み及び命令などのネットワークモデルの実行プロセスにおける関連データを保存するために使用されることができる。当該第１メモリ２００は、キャッシュなどの揮発性メモリのような内部メモリであることができる。当該第２メモリ３００は、ネットワークモデルに対応するオフラインモデルを保存するために使用されることができ、例えば、第２メモリ３００は、不揮発性メモリであることができる。

【００３２】

上記コンピューターデバイスの動作原理は、以下のモデル変換方法における各ステップの実行プロセスと一致し、当該コンピューターデバイスのプロセッサ１００がメモリ２００におけるコンピュータプログラムを実行すると、モデル変換方法における各ステップを実現し、具体的には、以下の説明を参照することができる。

【００３３】

もちろん、他の実施例において、当該コンピューターデバイスは、プロセッサと一つのメモリをさらに含み、当該コンピューターデバイスは、プロセッサが当該プロセッサに接続されたメモリを含むことができる。当該プロセッサは、図２～４に示したプロセッサを使用することができ、その具体的な構造は、上記のプロセッサ１００に関する説明を参照することができる。当該メモリは、複数の記憶ユニットを含むことができ、例えば、当該メモリは、第１記憶ユニット、第２記憶ユニット及び第３記憶ユニットを含むことができ、ここで、当該第１記憶ユニットは、コンピュータプログラムを保存するために使用されることができ、当該コンピュータプログラムは、本出願の実施例で提供されるモデル変換方法を実現するために使用される。当該第２記憶ユニットは、オリジナルネットワーク実行プロセスにおける関連データを保存するために使用されることができ、当該第３記憶ユニットは、オリジナルネットワークに対応するオフラインモデル及び当該オフラインモデルに対応するターゲットオフラインモデル及び予め設定されたモデル変換ルールなどを保存するために使用されることができる。さらに、当該メモリに含まれる記憶ユニットの数は、３より多いこともでき、ここでは具体的に限定しない。

【００３４】

図５に示したように、本出願の実施例は、上記のコンピューターデバイスに適用可能なモデル変換方法を提供する。当該モデル変換方法は、コンピューターデバイスが受信した初期オフラインモデルのモデル属性情報と当該コンピューターデバイスのハードウェア属

性情報が一致しない場合、当該コンピューターデバイスが当該初期オフラインモデルを実行することができるように、当該初期オフラインモデルを当該コンピューターデバイスのハードウェア属性情報と一致するターゲットオフラインモデルに変換する。具体的に、上記方法は、以下のステップを含むことができる。

【００３５】

Ｓ１００において、初期オフラインモデル及びコンピューターデバイスのハードウェア属性情報を獲得する。

【００３６】

具体的に、上記の初期オフラインモデルとは、コンピューターデバイスが直接に獲得したオフラインモデルを指し、当該初期オフラインモデルは、第２メモリ３００に保存されることができる。ここで、オフラインモデルは、オリジナルネットワークにおける計算ノードのネットワークの重み及び命令などの必要なネットワーク構造情報を含むことができ、命令は、当該計算ノードがどのような計算機能を実行するかを示すために使用され、具体的に当該オリジナルネットワークにおける各計算ノードの計算属性及び各計算ノードの間の接続関係などの情報を含むことができる。当該オリジナルネットワークは、ニューラルネットワークモデルなどのネットワークモデルであることができ、図９を参照することができる。従って、コンピューターデバイスは、当該オリジナルネットワークに対応するオフラインモデルを実行することによって、当該オリジナルネットワークの演算機能を実現することができ、同じオリジナルネットワークを再びコンパイルする必要がないため、当該ネットワークを実行するプロセッサの実行時間を短縮し、さらにプロセッサの処理速度及び効率を向上させる。選択的に、本出願の実施例における初期オフラインモデルは、オリジナルネットワークに従って直接に生成されたオフラインモデルであることができ、他のモデル属性情報を有するオフラインモデルの一回または複数回の変換後に獲得されたオフラインモデルであることもでき、ここでは具体的に限定しない。

【００３７】

Ｓ２００において、初期オフラインモデルとコンピューターデバイスのハードウェア属性情報に従って、初期オフラインモデルのモデル属性情報とコンピューターデバイスのハードウェア属性情報が一致するかどうかを判断する。

【００３８】

具体的に、当該初期オフラインモデルのモデル属性情報は、当該初期オフラインモデルのデータ構造及びデータタイプなどを含むことができる。例えば、初期オフラインモデルにおける各ネットワークの重み及び命令の配置方法、各ネットワークの重みのタイプ及び各命令のタイプなどである。上記のコンピューターデバイスのハードウェア属性情報は、当該コンピューターデバイスのモデル番号、当該コンピューターデバイスが処理できるデータタイプ（例えば、固定小数点と浮動小数点など）及びデータ構造などを含む。コンピューターデバイスは、上記初期オフラインモデルのモデル属性情報とコンピューターデバイスのハードウェア属性情報に従って、当該コンピューターデバイスが上記初期オフラインモデルの実行をサポートできるかどうかを判断して（即ち、当該コンピューターデバイスが上記の初期オフラインモデルを実行できるかどうか）、当該初期オフラインモデルのモデル属性情報とコンピューターデバイスのハードウェア属性情報が一致するかどうかを決定する。

【００３９】

選択的に、当該コンピューターデバイスが上記初期オフラインモデルの実行をサポートできる場合、当該初期オフラインモデルのモデル属性情報とコンピューターデバイスのハードウェア属性情報が一致すると判定する。当該コンピューターデバイスが上記初期オフラインモデルの実行をサポートしない場合、当該初期オフラインモデルのモデル属性情報と当該コンピューターデバイスのハードウェア属性情報が一致しないと判定する。

【００４０】

例えば、当該コンピューターデバイスは、そのハードウェア属性情報に従って、当該コンピューターデバイスが処理できるデータタイプ及びデータ構造を決定することができ、

当該初期オフラインモデルのモデル属性情報に従って、当該初期オフラインモデルのデータタイプ及びデータ構造を決定する。当該コンピューターデバイスが自身のハードウェア属性情報及び上記初期オフラインモデルのモデル属性情報に従って、当該コンピューターデバイスが上記初期オフラインモデルのデータタイプ及びデータ構造をサポートできると決定する場合、当該初期オフラインモデルのモデル属性情報と当該コンピューターデバイスのハードウェア属性情報が一致すると判定することができる。当該コンピューターデバイスが自身のハードウェア属性情報及び上記の初期オフラインモデルのモデル属性情報に従って、当該コンピューターデバイスが上記初期オフラインモデルのデータタイプ及びデータ構造をサポートしないと決定する場合、当該初期オフラインモデルのモデル属性情報と当該コンピューターデバイスのハードウェア属性情報が一致しないと判定することができる。

【００４１】

初期オフラインモデルのモデル属性情報とコンピューターデバイスのハードウェア属性情報が一致しない場合、ステップＳ３００を実行する：コンピューターデバイスのハードウェア属性情報と予め設定されたモデル変換ルールに従って、初期オフラインモデルをコンピューターデバイスのハードウェア属性情報に一致するターゲットオフラインモデルに変換する。

【００４２】

具体的に、当該初期オフラインモデルのモデル属性情報と当該コンピューターデバイスのハードウェア属性情報が一致しない場合、当該コンピューターデバイスが当該初期オフラインモデルを使用して対応する演算を実現するように、当該コンピューターデバイスが当該初期オフラインモデルの実行をサポートできないことを示し、このとき、当該初期オフラインモデルに変換処理を行う必要がある。即ち、当該初期オフラインモデルのモデル属性情報とコンピューターデバイスのハードウェア属性情報が一致しない場合、コンピューターデバイスのハードウェア属性情報と予め設定されたモデル変換ルールに従って、上記の初期オフラインモデルを当該コンピューターデバイスのハードウェア属性情報と一致するターゲットオフラインモデルに変換することができる。ここで、当該ターゲットオフラインモデルのモデル属性情報は当該コンピューターデバイスのハードウェア属性情報と一致し、当該コンピューターデバイスは、当該ターゲットオフラインモデルの実行をサポートすることができる。上記のように、当該ターゲットオフラインモデルは、オリジナルネットワークにおける各計算ノードのネットワークの重み及び命令などの必要なネットワーク構造情報も含む。

【００４３】

選択的に、上記の予め設定されたモデル変換ルールは、コンピューターデバイスの第１メモリまたは第２メモリに事前に保存されることができる。初期オフラインモデルのモデル属性情報とコンピューターデバイスのハードウェア属性情報が一致しない場合、初期オフラインモデルをターゲットオフラインモデルに変換するために、コンピューターデバイスは、第１メモリまたは第２メモリから対応するモデル変換ルールを獲得することができる。選択的に、上記予め設定されたモデル変換ルールは、一つ以上であることができ、一つ以上のモデル変換ルールは、上記の初期オフラインモデル及びターゲットオフラインモデルと１対１に対応するように保存されることができる。例えば、上記の初期オフラインモデル、ターゲットオフラインモデル及び予め設定されたモデル変換ルールは、マッピングテーブルの方法によって、対応して保存されることができる。選択的に、初期オフラインモデルのモデル属性情報、ターゲットオフラインモデルのモデル属性情報及び使用可能なモデル変換ルールの三つの間は１対１に対応するように保存される。

【００４４】

初期オフラインモデルのモデル属性情報とコンピューターデバイスのハードウェア属性情報が一致する場合、ステップＳ４００を実行することができる：コンピューターデバイスは、受信した初期オフラインモデルを直接に実行することができ、即ち、コンピューターデバイスは、オリジナルネットワークの演算機能を実現するために、当該初期オフライ

ンモデルに含まれるネットワークの重み及び命令に従って演算を実行することができる。具体的に、当該初期オフラインモデルのモデル属性情報と当該コンピューターデバイスのハードウェア属性情報が一致する場合、当該コンピューターデバイスが当該初期オフラインモデルの実行をサポートできることを示し、このとき、当該初期オフラインモデルに変換処理を行う必要がなく、コンピューターデバイスは、当該初期オフラインモデルを実行することができる。本実施例において、当該初期オフラインモデルを直接に実行するとは、当該初期オフラインモデルを使用して当該オリジナルネットワークに対応する機械ラーニングアルゴリズム（例えば、ニューラルネットワークアルゴリズム）を実行し、フォワード演算を実行することによりアルゴリズムのターゲットアプリケーション（例えば、音声認識などの人工知能アプリケーション）を実現することを指す。

【００４５】

本出願の実施例におけるモデル変換方法では、コンピューターデバイスは一つの初期オフラインモデルを受信するだけでよく、予め設定されたモデル変換ルールと当該コンピューターデバイスのハードウェア属性情報に従って、当該初期オフラインモデルをターゲットオフラインモデルに変換することができ、複数の異なるモデルデータを獲得する必要なく、コンピューターデバイスのデータ入力量を大幅に削減し、過剰なデータ入力量に起因するコンピューターデバイスのストレージ容量を超えるなどの問題を回避し、コンピューターデバイスの正常実行を保証する。同時に、上記モデル変換方法は、当該コンピューターデバイスのデータ処理量を削減することができ、さらに処理効率を向上させることができ、消費電力を削減する。同時に、上記のモデル変換プロセスにおいて、人間の介入が必要なく、自動化の程度が高く、使用が便利である。

【００４６】

選択的に、図６に示したように、上記のステップＳ３００は、以下のステップを含むことができる。

【００４７】

Ｓ３１０において、コンピューターデバイスのハードウェア属性情報に従って、ターゲットオフラインモデルのモデル属性情報を決定する。

【００４８】

具体的に、当該コンピューターデバイスは、自身のハードウェア属性情報に従って、当該コンピューターデバイスがサポートできるデータタイプ及びデータ構造などを決定することができるため、ターゲットオフラインモデルのデータタイプ及びデータ構造などのモデル属性情報を決定することができる。即ち、当該コンピューターデバイスは、そのハードウェア属性情報に従ってターゲットオフラインモデルのモデル属性情報を決定することができ、当該ターゲットオフラインモデルのモデル属性情報は、ターゲットオフラインモデルのデータ構造及びデータタイプなどの情報を含むことができる。

【００４９】

Ｓ３２０において、初期オフラインモデルのモデル属性情報とターゲットオフラインモデルのモデル属性情報に従って、複数の予め設定されたモデル変換ルールから一つのモデル変換ルールをターゲットモデル変換ルールとして選択する。

【００５０】

具体的に、初期オフラインモデル、ターゲットオフラインモデルと予め設定されたモデル変換ルールに１対１に対応するマッピング関係が存在するため、当該初期オフラインモデルのモデル属性情報とターゲットオフラインモデルのモデル属性情報に従って、ターゲットモデル変換ルールを決定することができる。ここで、当該モデル変換ルールは、データタイプの変換方法及びデータ構造の変換方法などを含むことができる。

【００５１】

Ｓ３３０において、初期オフラインモデルのモデル属性情報とターゲットモデル変換ルールに従って、初期オフラインモデルをターゲットオフラインモデルに変換する。

【００５２】

具体的に、コンピューターデバイスは、当該ターゲットモデル変換ルールによって提供

される変換方法に従って、初期オフラインモデルをターゲットオフラインモデルに変換して、当該コンピューターデバイスが当該ターゲットオフラインモデルを実行して演算を実行できるようにする。

【００５３】

選択的に、初期オフラインモデルとターゲットオフラインモデルとの間に複数の使用可能なモデル変換ルールが存在する場合、コンピューターデバイスは、予めに設定されたアルゴリズムに従ってターゲットオフラインモデルを自動的に選択することができる。具体的に、上記のステップＳ３２０は、以下のステップをさらに含む。

【００５４】

初期オフラインモデルのモデル属性情報とターゲットオフラインモデルのモデル属性情報に従って、複数の予め設定されたモデル変換ルールから一つ以上の使用可能なモデル変換ルールを選択する。

【００５５】

一つ以上の使用可能なモデル変換ルールに優先順位を付け、優先順位の高い使用可能なモデル変換ルールを前記ターゲット変換ルールとして使用する。

【００５６】

具体的に、上記の使用可能なモデル変換ルールとは、初期オフラインモデルをターゲットオフラインモデルに変換する変換ルールを指す。一つ以上の使用可能なモデル変換ルールが存在する場合、初期オフラインモデルをターゲットオフラインモデルに変換する方法には、多数の異なる方法があることを示す。上記の優先順位を付ける方法は、予め設定されることができ、ユーザーによって定義されることもできる。

【００５７】

選択的に、コンピューターデバイスは、各使用可能なモデル変換ルールに対応するプロセスパラメーターを獲得することができ、各使用可能なモデル変換ルールに対応するプロセスパラメーターに従って、一つ以上の使用可能なモデル変換ルールに優先順位を付ける。ここで、当該プロセスパラメーターは、当該使用可能なモデル変換ルールを使用して、初期オフラインモデルをターゲットオフラインモデルに変換する場合に関与するコンピューターデバイスのパフォーマンスパラメータであることができる。選択的に、当該プロセスパラメーターは、変換速度、消費電力、メモリ使用率及びディスクＩ／Ｏ占有率中の一つまたは複数を含むことができる。

【００５８】

例えば、コンピューターデバイスは、初期オフラインモデルをターゲットオフラインモデルに変換するプロセスにおける変換速度、変換消費電力、メモリ使用率及びディスクＩ／Ｏ占有率などの一つまたは複数プロセスパラメーターの組合せに従って、上記の各使用可能なモデル変換ルールに対してスコアリングし（例えば、スコア値を取得するために、各参照要因に加重計算を行う）、最高スコアの使用可能なモデル変換ルールを最高優先順位の使用可能なモデル変換ルールとして使用することができる。即ち、最高スコアの使用可能なモデル変換ルールを当該ターゲット変換ルールとして使用することができる。その後、上記ステップＳ３３０を実行することができ、初期オフラインモデルのモデル属性情報とターゲットモデル変換ルールに従って、初期オフラインモデルをターゲットオフラインモデルに変換する。このようにして、コンピューターデバイスの処理速度及び効率を向上させるために、モデル変換プロセス中のデバイスのパフォーマンス要因を合わせて、より優れたモデル変換ルールを選択することができる。

【００５９】

選択的に、図３に示したように、上記方法は、以下のをステップをさらに含む。

【００６０】

Ｓ５００において、ターゲットオフラインモデルをコンピューターデバイスの第１メモリまたは第２メモリに保存する。

【００６１】

具体的に、コンピューターデバイスは、獲得されたターゲットオフラインモデルを当該

コンピューターデバイスのローカルメモリ（例えば、第１メモリ）に保存することができる。選択的に、コンピューターデバイスは、さらに獲得されたターゲットオフラインモデルを当該コンピューターデバイスに接続された外部メモリ（例えば、第２メモリ）に保存することができ、当該外部メモリは、クラウドストレージまたは他のメモリなどであることができる。このようにして、当該コンピューターデバイスが当該ターゲットオフラインモデルを繰り返して使用する必要がある場合、上記の変換操作を繰り返して実行する必要なく、当該ターゲットオフラインモデルを直接に読み取ることで、対応する演算を実現することができる。

【００６２】

選択的に、コンピューターデバイスが上記のターゲットオフラインモデルを実行するプロセスは、以下のステップを含むことができる。

【００６３】

オリジナルネットワークを実行するために、ターゲットオフラインモデルを獲得し、ターゲットオフラインモデルを実行する。ここで、ターゲットオフラインモデルは、オリジナルネットワークにおける各オフラインノードに対応するネットワークの重み、命令及び各オフラインノードとオリジナルネットワーク中の他の計算ノードとの間のインターフェースデータなどの必要なネットワーク構造情報を含む。

【００６４】

具体的に、コンピューターデバイスが当該ターゲットオフラインモデルを使用して演算を実現する必要がある場合、第１メモリまたは第２メモリから当該ターゲットオフラインモデルを直接に獲得することができ、ターゲットオフラインモデル中のネットワークの重み及び命令などに従って演算を行って、オリジナルネットワークの演算機能を実現する。このようにして、当該コンピューターデバイスが当該ターゲットオフラインモデルを繰り返して使用する必要がある場合、上記の変換操作を繰り返す必要なく、当該ターゲットオフラインモデルを直接に読み取るだけで、対応する演算を実現することができる。

【００６５】

本実施例において、当該ターゲットオフラインモデルを直接に実行するとは、当該初期ターゲットオフラインモデルを使用して当該オリジナルネットワークに対応する機械ラーニングアルゴリズム（例えば、ニューラルネットワークアルゴリズム）を実行し、フォワード演算を実行することによりアルゴリズムのターゲットアプリケーション（例えば、音声認識などの人工知能アプリケーション）を実現することを指す。

【００６６】

選択的に、当該コンピューターデバイスには、アプリケーションソフトウェア（Ａｐｐｌｉｃａｔｉｏｎ）がインストールされ、コンピューターデバイス上のアプリケーションソフトウェアを介して、初期オフラインモデルを獲得することができ、当該アプリケーションソフトウェア（Ａｐｐｌｉｃａｔｉｏｎ）は、メモリ２００または外部メモリから初期オフラインモデルを読み取るためのインターフェイスを提供することができるため、当該アプリケーションソフトウェアを介して当該初期オフラインモデルを獲得することができる。選択的に、当該アプリケーションソフトウェアは、コンピューターデバイスのハードウェア属性情報を読み取るためのインターフェイスをさらに提供するため、当該アプリケーションソフトウェアを介して当該コンピューターデバイスのハードウェア属性情報を獲得することができる。選択的に、当該アプリケーションソフトウェアは、予め設定されたモデルルールを読み取るためのインターフェイスをさらに提供することができるため、当該アプリケーションソフトウェアを介して予め設定されたモデル変換ルールを獲得することができる。

【００６７】

もちろん、他の実施例において、当該コンピューターデバイスは、入力／出力インターフェイス（例えば、Ｉ／Ｏインターフェイス）などをさらに提供することができ、当該入力／出力インターフェイスは、初期オフラインモデルの獲得、またはターゲットオフラインモデルの出力のために使用される。当該コンピューターデバイスのハードウェア属性情

報は、当該コンピューターデバイスに事前に保存されることができる。

【００６８】

一つの実施例において、図７に示したように、本出願の実施例は、オリジナルネットワークモデルに従って、オフラインモデルを生成する方法をさらに提供し、獲得されたオリジナルネットワークの関連データに従って、当該オリジナルネットワークのオフラインモデルを生成及び保存するために使用されるため、プロセッサが当該オリジナルネットワークを再び実行する場合、同じオリジナルネットワークを再びコンパイルする必要がなく、当該オリジナルネットワークに対応するオフラインモデルを直接に実行することができるため、当該ネットワークを実行するプロセッサの実行時間を短縮し、さらにプロセッサの処理速度及び効率を向上させる。ここで、オリジナルネットワークは、図９に示したようなネットワークのようなニューラルネットワークまたは非ニューラルネットワークなどのネットワークモデルであることができる。具体的に、上記方法は、以下のステップを含む。

【００６９】

Ｓ０１０において、オリジナルネットワークのモデルデータセット及びモデル構造パラメータを獲得する。

【００７０】

具体的に、コンピューターデバイスのプロセッサは、オリジナルネットワークのモデルデータセット及びモデル構造パラメータを獲得することができ、当該オリジナルネットワークのモデルデータセット及びモデル構造パラメータを介して、当該オリジナルネットワークのネットワーク構造図を獲得することができる。ここで、モデルデータセットは、オリジナルネットワークにおける各計算ノードに対応するネットワークの重みなどのデータを含み、図９に示したニューラルネットワークにおけるＷ１～Ｗ６は、計算ノードのネットワークの重みを表すために使用される。モデル構造パラメータは、オリジナルネットワークにおける複数の各計算ノードの接続関係及び各計算ノードの計算属性を含み、ここで、計算ノードの間の接続関係は、計算ノードの間にデータ伝達があるかどうかを表すために使用され、例えば、複数の各計算ノードの間にデータフローの伝達がある場合、複数の計算ノードの間に接続関係があると説明することができる。さらに、計算ノードの接続関係は、入力関係及び出力関係などを含むことができる。図９に示したように、計算ノードＦ１が計算ノードＦ４及びＦ５の入力として出力する場合、計算ノードＦ１と計算ノードＦ４との間に接続関係があり、計算ノードＦ１と計算ノードＦ４との間に接続関係があると説明できる。別の例として、計算ノードＦ１と計算ノードＦ２との間にデータ伝達がない場合、計算ノードＦ１と計算ノードＦ２との間に接続関係が存在しないと説明できる。

【００７１】

各計算ノードの計算属性は、対応する計算ノードの計算タイプ及び計算パラメータを含むことができ、ここで、計算ノードの計算タイプとは、当該計算ノードがある計算を完了することに使用されることを指し、例えば、各計算ノードの計算タイプは、加算、減算、及び畳み込み算など含むことができ、対応的に、当該計算ノードは、加算を実現するための計算ノード、減算を実現するための各計算ノードまたは畳み込み算を実現するための計算ノードなどであることができる。計算ノードの計算パラメータは、当該計算ノードに対応する計算タイプを完了する必要なパラメータであることができる。例えば、計算ノードの計算タイプは、加算を実現するための計算ノードであることができ、対応的に、当該計算ノードの計算パラメータは、加算における加数であることができ、当該加算における被加数は、入力データとして獲得モジュールを介して獲得することができ、または、当該加算における被加数は、当該計算ノードの前の各計算ノードの出力データなどであることができる。

【００７２】

Ｓ０２０において、オリジナルネットワークのモデルデータセット及びモデル構造パラメータに従ってオリジナルネットワークを実行して、オリジナルネットワークにおける各計算ノードに対応する命令を獲得する。

【００７３】
具体的に、コンピューターデバイスのプロセッサは、オリジナルネットワークのモデルデータセット及びモデル構造パラメータに従って当該オリジナルネットワークを実行して、オリジナルネットワークにおける各計算ノードに対応する命令を獲得することができる。さらに、プロセッサは、当該オリジナルネットワークの入力データをさらに獲得することができ、オリジナルネットワークの入力データ、ネットワークモデルデータセット及びモデル構造パラメータに従ってオリジナルネットワークを実行して、当該オリジナルネットワークにおける各計算ノードに対応する命令を獲得することができる。さらに、上記当該オリジナルネットワークを実行して各計算ノードの命令を獲得する過程は、実際にコンパイルの過程であり、当該コンパイル過程は、コンピュータシステムのプロセッサまたは仮想デバイスを介して実現することができる。即ち、コンピュータシステムのプロセッサまたは仮想デバイスは、オリジナルネットワークのモデルデータセット及びモデル構造パラメータに従ってオリジナルネットワークを実行する。ここで、仮想デバイスとは、メモリのメモリ空間でプロセッサ実行空間のセクションを仮想することを指す。
【００７４】
明らかに、本実施例におけるオリジナルネットワークの実行とは、プロセッサが人工ニューラルネットワークモデルデータを使用して、ある機械ラーニングアルゴリズム（例えば、ニューラルネットワークアルゴリズム）を実行し、フォワード演算を実行することにより、アルゴリズムのターゲットアプリケーション（例えば、音声認識などの人工知能アプリケーション）を実現することを指す。
【００７５】
Ｓ０３０において、オリジナルネットワークの各計算ノードに対応するネットワークの重み及び命令に従って、オリジナルネットワークに対応するオフラインモデルを生成し、オリジナルネットワークに対応するオフラインモデルを不揮発性メモリ（データベース）に保存する。
【００７６】
具体的に、当該プロセッサの制御モジュールは、オリジナルネットワークの各計算ノードに対応するネットワークの重み及び命令に従って、当該オリジナルネットワークに対応するオフラインモデルを生成することができ、例えば、当該プロセッサの制御モジュールは、オリジナルネットワークの各計算ノードに対応するネットワークの重み及び命令を不揮発性第２メモリに保存して、オフラインモデルの生成及び保存を実現することができる。ここで、オリジナルネットワークの各計算ノードについて、当該計算ノードのネットワークの重みと命令は、１対１に対応するように保存される。このようにして、当該オリジナルネットワークを再び実行する場合、当該オリジナルネットワークの各計算ノードに対してオンラインでコンパイルして命令を獲得することなく、不揮発性メモリから当該オリジナルネットワークに対応するオフラインモデルを直接に獲得することができ、それに対応するオフラインモデルに従ってオリジナルネットワークを実行し、システムの実行速度及び効率を向上させる。
【００７７】
明らかに、本実施例において、当該オリジナルネットワークに対応するオフラインモデルを直接に実行するとは、オフラインモデルを使用して当該オリジナルネットワークに対応する機械ラーニングアルゴリズム（例えば、ニューラルネットワークアルゴリズム）実行し、フォワード演算を実行することによりアルゴリズムのターゲットアプリケーション（例えば、音声認識などの人工知能アプリケーション）を実現することを指す。
【００７８】
選択的に、図８に示したように、前記ステップＳ２００は、以下のステップを含むことができる。
【００７９】
Ｓ０２１において、オリジナルネットワークのモデル構造パラメータに従って、オリジナルネットワークにおける各計算ノードの実行順序を獲得する。

【００８０】
具体的に、プロセッサは、オリジナルネットワークのモデル構造パラメータに従って、オリジナルネットワークにおける各計算ノードの実行順序を獲得することができ、さらに、プロセッサの演算モジュールは、オリジナルネットワークにおける各計算ノードの接続関係に従って、オリジナルネットワークにおける各計算ノードの実行順序を獲得することができる。例えば、図９に示したように、計算ノードＦ４の入力データは、計算ノードＦ１の出力データ及び計算ノードＦ２の出力データであり、計算ノードＦ６の入力データは、計算ノードＦ４の出力データ及び計算ノードＦ５の出力データである。従って、図９に示したニューラルネットワークにおける各計算ノードの実行順序は、Ｆ１－Ｆ２－Ｆ３－Ｆ４－Ｆ５－Ｆ６またはＦ１－Ｆ３－Ｆ２－Ｆ５－Ｆ４－Ｆ６などであることができる。もちろん、計算ノードＦ１、Ｆ２及びＦ３は、並列に実行することができ、計算ノードＦ４及びＦ５も並列に実行することができ、ここでは単に例として説明し、実行順序は具体的に限定されない。

【００８１】
Ｓ０２２において、オリジナルネットワークにおける各計算ノードの実行順序に従ってオリジナルネットワークを実行して、オリジナルネットワークにおける各計算ノードに対応する命令をそれぞれ獲得する。

【００８２】
具体的に、プロセッサは、オリジナルネットワークにおける各計算ノードの実行順序に従って当該オリジナルネットワークを実行して、オリジナルネットワークにおける各計算ノードに対応する命令を獲得することができ、即ち、プロセッサは、オリジナルネットワークのモデルデータセットなどのデータをコンパイルして各計算ノードに対応する命令を獲得することができ、各計算ノードに対応する命令を介して当該各計算ノードがどのようなコンピューティング機能を実現するかを知ることができ、即ち、当該各計算ノードの計算タイプ及び計算パラメータなどの計算属性を獲得することができる。

【００８３】
さらに、図８に示したように、前記ステップＳ３００はさらに以下のステップを含む。

【００８４】
Ｓ０３１において、オリジナルネットワークのモデルデータセット及びモデル構造パラメータに従って、オリジナルネットワークのメモリ割り当て方法を獲得する。

【００８５】
具体的に、プロセッサは、オリジナルネットワークのモデルデータセット及びモデル構造パラメータに従って、オリジナルネットワークのメモリ割り当て方法を獲得することができ、オリジナルネットワークのモデル構造パラメータに従って、オリジナルネットワークにおける各計算ノードの実行順序を獲得することができ、オリジナルネットワークにおける各計算ノードの実行順序に従って、現在のネットワークのメモリ割り当て方法を決定する。例えば、各計算ノードの実行順序に従って、各計算ノードの実行過程中の関連データは、一つのスタックに保存される。ここで、メモリ割り当て方法とは、オリジナルネットワークにおける各計算ノードに関連するデータ（入力データ、出力データ、ネットワークの重みデータ及び中間結果データなどを含む）がメモリ空間（例えば、第１メモリ）での保存位置を決定することを指す。例えば、データテーブルを使用して各計算ノードに関連するデータ（入力データ、出力データ、ネットワークの重みデータ及び中間結果データなど）とメモリ空間のマッピング関係を保存することができる。

【００８６】
Ｓ０３２において、オリジナルネットワークのメモリ割り当て方法に従って、オリジナルネットワークの実行過程中の関連データを第１メモリに保存する。ここで、オリジナルネットワークの実行過程中の関連データは、オリジナルネットワークの各計算ノードに対応するネットワークの重み、命令、入力データ、中間計算結果及び出力データなどを含む。例えば、図９に示したように、Ｘ１とＸ２は、当該ニューラルネットワークの入力データを表し、Ｙは、当該ニューラルネットワークの出力データを表し、プロセッサは、当該

ニューラルネットワークの出力データをロボットまたは異なるデジタルインターフェースを制御する制御命令に転換することができる。Ｗ１～Ｗ６は、計算ノードＦ１、Ｆ２及びＦ３に対応するネットワークの重みを表すために使用され、計算ノードＦ１～Ｆ５の出力データは、中間計算結果として使用することができる。プロセッサは、決定されたメモリ割り当て方法に従って、オリジナルネットワークの実行過程中の関連データを内部メモリまたはキャッシュなどの揮発性メモリのような第１メモリに保存することができ、具体的な保存方法は、図１０における左半部の保存空間を参照することができる。

【００８７】

Ｓ０３３において、第１メモリからオリジナルネットワークにおける各計算ノードに対応するネットワークの重み及び命令を獲得し、オリジナルネットワークにおける各計算ノードに対応するネットワークの重み及び命令を第２メモリに保存し、オフラインモデルを生成する。ここで、第２メモリは、外部メモリなどの不揮発性メモリであることができる。当該オフラインモデルの生成過程は、具体的に図１０を参照することができ、図１０における右半部の保存空間に保存されたのは、オリジナルネットワークの対応するオフラインモデルである。

【００８８】

図９及び図１０に示したように、以下、図面に合わせて、上記オフラインモデルの生成過程を説明する。

【００８９】

まず、図９に示したように、プロセッサは、当該オリジナルネットワークのモデルデータセット、モデル構造パラメータ及び入力データを獲得することができるため、当該オリジナルネットワークのモデルデータセット及びモデル構造パラメータに従って当該オリジナルネットワークのネットワーク構造図を獲得することができる。

【００９０】

次に、プロセッサは、オリジナルネットワークのモデル構造パラメータに従って、オリジナルネットワークの各計算ノードの接続関係を獲得することができ、各計算ノードの接続関係に従ってオリジナルネットワークにおける各計算ノードの実行順序、及びオリジナルネットワークの実行過程中のメモリ割り当て方法を獲得するため、オリジナルネットワークの実行過程中の関連データの保存位置を獲得することができる。図１０の左半部の保存空間に示したように、オリジナルネットワークの実行過程中の関連データは、各計算ノード実行順序に従って一つのスタックに保存されることができる。

【００９１】

最後に、プロセッサは、オリジナルネットワークの各計算ノードに対応するネットワークの重み及び命令を不揮発性の第２メモリに保存し、オフラインモデルを生成することができ、当該オフラインモデルの保存方法は、図８における右半部の保存空間を参照することができる。また、当該オフラインモデルは、当該オリジナルネットワークを実行するに必要なネットワークの重み及び命令などのデータのみを含み、オリジナルネットワークの実行過程中の入力データ、出力データまたは中間計算結果などを保存する必要がないため、第２メモリにおける保存空間の消費を減少することができる。

【００９２】

さらなる改善として、オフラインモデルには、ノードインターフェースデータがさらに含まれ、ノードインターフェースデータは、オリジナルネットワークの各計算ノードの接続関係を表すために使用される。具体的に、ノードインターフェースデータは、各計算ノードの入力データソース及び出力データソースを含むことができる。例えば、図９に示したように、ノードインターフェースデータは、計算ノードＦ１、Ｆ２及びＦ３を開始各計算ノードとして含むことができ、それぞれ予め設定された入力データを入力し、計算ノードＦ１の出力データは、計算ノードＦ４及び計算ノードＦ５の入力データなどである。このようにして、当該オリジナルネットワークを再び実行する場合、オリジナルネットワークの開始各計算ノードと入力データのみを獲得し、その後、当該オリジナルネットワークに対応するオフラインモデルに従って当該オリジナルネットワークを実行することができ

る。

【００９３】

図５～８のフロチャートにおける各ステップは矢印で示されるように順次的に表示されるが、これらのステップは必ずしも矢印で示される順序で実行されるとは限らないことを理解されたい。本明細書で明示的に説明される場合を除き、これらのステップの実行は厳密に制限されておらず、これらのステップは他の順序で実行されてもよい。さらに、図５～８における少なくとも一部のステップは、複数のサブステップまたは複数の段階を含むことができ、これらのサブステップまたは段階は、必ずしも同時に実行される必要なく、異なる時間に実行されることができ、これらのサブステップまたは段階の実行順序も必ずしも順次に実行される必要なく、他のステップまたは他のステップのサブステップまたは段階の少なくとも一部と交代にまたは交互に実行されることができる。

【００９４】

当業者は、上記実施例方法における全部または一部のプロセスの実現は、コンピュータプログラムに介して関連するハードウェアを命令して完了し、前記コンピュータプログラムは、不揮発性コンピュータ可読記憶媒体に保存されることができ、当該プログラムは実行時に、上記各方法の実施例のプロセスを含むことができることを理解できる。ここで、本出願で提供される各実施例に使用されるメモリ、ストレージ、データベースまたは他の媒体への参照は、すべて不揮発性及び／または揮発性メモリを含むことができる。不揮発性メモリは、読み取り専用メモリ（ＲＯＭ）、プログラマブルＲＯＭ（ＰＲＯＭ）、電気的にプログラム可能なＲＯＭ（ＥＰＲＯＭ）、電気的に消去可能なプログラム可能なＲＯＭ（ＥＥＰＲＯＭ）またはフラッシュメモリを含むことができる。揮発性メモリは、ランダムアクセスメモリ（ＲＡＭ）または外部高速キャッシュメモリを含むことができる。制限ではない説明として、ＲＡＭは、スタティックＲＡＭ（ＳＲＡＭ）、ダイナミックＲＡＭ（ＤＲＡＭ）、同期ＤＲＡＭ（ＳＤＲＡＭ）、ダブルデータレートＳＤＲＡＭ（ＤＤＲＳＤＲＡＭ）、拡張ＳＤＲＡＭ（ＥＳＤＲＡＭ）、同期リンク（Ｓｙｎｃｈｌｉｎｋ）、ＤＲＡＭ（ＳＬＤＲＡＭ）、ラムバス（Ｒａｍｂｕｓ）ダイレクトＲＡＭ（ＲＤＲＡＭ）、ダイレクトラムバスダイナミックＲＡＭ（ＤＲＤＲＡＭ）、ラムバスダイナミックＲＡＭ（ＲＤＲＡＭ）などのようなさまざまな形式で獲得することができる。

【００９５】

本出願の実施例は、獲得モジュール、判断モジュール及び変換モジュールを含むモデル変換装置をさらに提供する。ここで、獲得モジュールは、初期オフラインモデルとコンピューターデバイスのハードウェア属性情報を獲得するために使用され、判断モジュールは、初期オフラインモデルとコンピューターデバイスのハードウェア属性情報に従って、初期オフラインモデルのモデル属性情報とコンピューターデバイスのハードウェア属性情報が一致するかどうかを判断するように使用され、変換モジュールは、初期オフラインモデルのモデル属性情報とコンピューターデバイスのハードウェア属性情報が一致しない場合、コンピューターデバイスのハードウェア属性情報と予め設定されたモデル変換ルールに従って、初期オフラインモデルを前記コンピューターデバイスのハードウェア属性情報と一致するターゲットオフラインモデルに変換するように使用される。

【００９６】

選択的に、当該モデル変換装置は、当該コンピューターデバイスにインストールされたアプリケーションソフトウェア（Ａｐｐｌｉｃａｔｉｏｎ）であることができる。当該アプリケーションソフトウェア（Ａｐｐｌｉｃａｔｉｏｎ）は、メモリ２００または外部メモリから初期オフラインモデルを読み取るためのインターフェイスを提供することができるため、当該アプリケーションソフトウェアを介して当該初期オフラインモデルを獲得することができる。選択的に、当該アプリケーションソフトウェアは、コンピューターデバイスのハードウェア属性情報を読み取るためのインターフェイスをさらに提供するため、当該アプリケーションソフトウェアを介して当該コンピューターデバイスのハードウェア属性情報を獲得することができる。選択的に、当該アプリケーションソフトウェアは、予め設定されたモデルルールを読み取るためのインターフェイスをさらに提供することがで

きるため、当該アプリケーションソフトウェアを介して予め設定されたモデル変換ルールを獲得することができる。さらに、当該アプリケーションソフトウェアは、初期オフラインモデルとコンピューターデバイスのハードウェア属性情報及び予め設定されたモデル変換ルールに従って、当該初期オフラインモデルのモデル属性情報とコンピューターデバイスのハードウェア属性情報が一致しない場合、当該初期オフラインモデルを当該コンピューターデバイスのハードウェア属性情報と一致するターゲットオフラインモデルに変換し、当該ターゲットオフラインモデルを第１メモリまたは第２メモリに保存する。

【００９７】

モデル変換装置に関する具体的な限定は、上記のモデル変換方法に対する限定を参照することができ、ここでは繰り返して説明しない。上記モデル変換装置中の各モジュールは、ソフトウェア、ハードウェア、及びそれらの組み合わせによって全体的または部分的に実現されることができる。プロセッサが上記のモジュールに対応する操作を実行するために呼び出されるように、上記各モジュールは、コンピューター装置のプロセッサに組み込まれるか、コンピューター装置のプロセッサーから独立していてもよく、ソフトウェア形式でコンピューター装置のメモリに保存されてもよい。

【００９８】

なお、本発明の一実施例は、コンピュータプログラムが保存されたコンピュータ可読記憶媒体をさらに提供し、前記コンピュータプログラムは、プロセッサによって実行される場合、請求項に記載のいずれかの一つの実施例の方法のステップを実現する。選択的に、当該コンピュータ可読記憶媒体は、不揮発性及び／または揮発性メモリを含むことができる。不揮発性メモリは、読み取り専用メモリ（ＲＯＭ）、プログラマブルＲＯＭ（ＰＲＯＭ）、電気的にプログラム可能なＲＯＭ（ＥＰＲＯＭ）、電気的に消去可能なプログラム可能なＲＯＭ（ＥＥＰＲＯＭ）またはフラッシュメモリを含むことができる。揮発性メモリは、ランダムアクセスメモリ（ＲＡＭ）または外部高速キャッシュメモリを含むことができる。制限ではない説明として、ＲＡＭは、スタティックＲＡＭ（ＳＲＡＭ）、ダイナミックＲＡＭ（ＤＲＡＭ）、同期ＤＲＡＭ（ＳＤＲＡＭ）、ダブルデータレートＳＤＲＡＭ（ＤＤＲＳＤＲＡＭ）、拡張ＳＤＲＡＭ（ＥＳＤＲＡＭ）、同期リンク（Ｓｙｎｃｈｌｉｎｋ）、ＤＲＡＭ（ＳＬＤＲＡＭ）、ラムバス（Ｒａｍｂｕｓ）ダイレクトＲＡＭ（ＲＤＲＡＭ）、ダイレクトラムバスダイナミックＲＡＭ（ＤＲＤＲＡＭ）、ラムバスダイナミックＲＡＭ（ＲＤＲＡＭ）などのようなさまざまな形式で獲得することができる。

【００９９】

具体的に、上記コンピュータプログラムがプロセッサによって実行される場合、以下のステップを実現する。

【０１００】

Ｓ１００において、初期オフラインモデルとコンピューターデバイスのハードウェア属性情報を獲得する。

【０１０１】

具体的に、上記の初期オフラインモデルとは、コンピューターデバイスが直接に獲得したオフラインモデルを指し、当該初期オフラインモデルは、第２メモリ３００に保存されることができる。ここで、オフラインモデルは、オリジナルネットワークにおける計算ノードのネットワークの重み及び命令などの必要なネットワーク構造情報を含むことができ、ここで、命令は、当該計算ノードがどのような計算機能を実行するかを示すために使用され、具体的に当該オリジナルネットワークにおける各計算ノードの計算属性及び各計算ノードの間の接続関係などの情報を含むことができる。選択的に、本出願の実施例における初期オフラインモデルは、オリジナルネットワークに従って直接に生成されたオフラインモデルであることができ、他のモデル属性情報を有するオフラインモデルの一回または複数回の変換後に獲得されたオフラインモデルであることもでき、ここでは具体的に限定しない。

【０１０２】

Ｓ２００において、初期オフラインモデルとコンピューターデバイスのハードウェア属

性情報に従って、初期オフラインモデルのモデル属性情報とコンピューターデバイスのハードウェア属性情報が一致するかどうかを判断する。

【０１０３】

具体的に、当該初期オフラインモデルのモデル属性情報は、当該初期オフラインモデルのデータ構造及びデータタイプなどを含むことができる。例えば、初期オフラインモデルにおける各ネットワークの重み及び命令の配置方法、各ネットワークの重みのタイプ及び各命令のタイプなどである。上記のコンピューターデバイスのハードウェア属性情報は、当該コンピューターデバイスのモデル番号、当該コンピューターデバイスが処理できるデータタイプ（例えば、固定小数点と浮動小数点など）及びデータ構造などを含む。

【０１０４】

例えば、当該コンピューターデバイスは、そのハードウェア属性情報に従って、当該コンピューターデバイスが処理できるデータタイプ及びデータ構造を決定することができ、当該初期オフラインモデルのモデル属性情報に従って、当該初期オフラインモデルのデータタイプ及びデータ構造を決定する。当該コンピューターデバイスが自身のハードウェア属性情報及び上記初期オフラインモデルのモデル属性情報に従って、当該コンピューターデバイスが上記初期オフラインモデルのデータタイプ及びデータ構造をサポートできると決定する場合、当該初期オフラインモデルのモデル属性情報と当該コンピューターデバイスのハードウェア属性情報が一致すると判定することができる。当該コンピューターデバイスが自身のハードウェア属性情報及び上記の初期オフラインモデルのモデル属性情報に従って、当該コンピューターデバイスが上記初期オフラインモデルのデータタイプ及びデータ構造をサポートしないと決定する場合、当該初期オフラインモデルのモデル属性情報と当該コンピューターデバイスのハードウェア属性情報が一致しないと判定することができる。

【０１０５】

初期オフラインモデルのモデル属性情報とコンピューターデバイスのハードウェア属性情報が一致しない場合、ステップＳ３００を実行する：コンピューターデバイスのハードウェア属性情報と予め設定されたモデル変換ルールに従って、初期オフラインモデルをコンピューターデバイスのハードウェア属性情報に一致するターゲットオフラインモデルに変換する。

【０１０６】

具体的に、当該初期オフラインモデルのモデル属性情報と当該コンピューターデバイスのハードウェア属性情報が一致しない場合、当該コンピューターデバイスが当該初期オフラインモデルを使用して対応する演算を実現するように、当該コンピューターデバイスが当該初期オフラインモデルの実行をサポートできないことを示し、このとき、当該初期オフラインモデルに変換処理を行う必要がある。即ち、当該初期オフラインモデルのモデル属性情報とコンピューターデバイスのハードウェア属性情報が一致しない場合、コンピューターデバイスのハードウェア属性情報と予め設定されたモデル変換ルールに従って、上記の初期オフラインモデルを当該コンピューターデバイスのハードウェア属性情報と一致するターゲットオフラインモデルに変換することができる。ここで、当該ターゲットオフラインモデルのモデル属性情報は当該コンピューターデバイスのハードウェア属性情報と一致し、当該コンピューターデバイスは、当該ターゲットオフラインモデルの実行をサポートすることができる。上記のように、当該ターゲットオフラインモデルは、オリジナルネットワークにおける各計算ノードのネットワークの重み及び命令などの必要なネットワーク構造情報も含む。

【０１０７】

初期オフラインモデルのモデル属性情報とコンピューターデバイスのハードウェア属性情報が一致する場合、ステップＳ４００を実行することができる：コンピューターデバイスは、受信した初期オフラインモデルを直接に実行することができ、即ち、コンピューターデバイスは、オリジナルネットワークの演算機能を実現するために、当該初期オフラインモデルに含まれるネットワークの重み及び命令に従って演算を実行することができる。

具体的に、当該初期オフラインモデルのモデル属性情報と当該コンピューターデバイスのハードウェア属性情報が一致する場合、当該コンピューターデバイスが当該初期オフラインモデルの実行をサポートできることを示し、このとき、当該初期オフラインモデルに変換処理を行う必要がなく、コンピューターデバイスは、当該初期オフラインモデルを実行することができる。

【０１０８】

プロセッサによって上記コンピュータプログラムを実行するプロセスは、上記実施例のモデル変換方法の実行プロセスと一致することは明らかであるはずであり、具体的には、上記の説明を参照することができ、ここでは繰り返して説明しない。

【０１０９】

上記各実施例の同一または類似の部分は相互に参照され、いくつかの実施例で詳細に説明されていない部分は他の実施例の同一または類似の内容を参照することができることが理解される。

【０１１０】

本出願の説明において、「第１」、「第２」などの用語は説明目的のみに使用され、相対的な重要性を示すまたは暗示するものとして解釈されるべきではないことに留意されたい。なお、本出願の説明において、「複数」の意味は、特に明記しない限り、少なくとも二つを意味する。

【０１１１】

フローチャートまたはここで他の方法で説明した任意のプロセスまたは方法は、特定の論理機能またはプロセスのステップを実現するための一つまたはさらに複数の実行可能な命令を含むコードのモジュール、セグメントまたは部分を表すと理解され得る。本出願の選択的な実施例の範囲には、追加の実現が含まれ、ここで、機能は、示されたまたは議論された順序ではなく、関与される機能に応じた実質的に同時の方法または反対の順序で実行されることができ、これは、本出願の実施例が属する技術分野の当業者によって理解されるべきである。

【０１１２】

本出願の各部分は、ハードウェア、ソフトウェア、ファームウェア、またはそれらの組み合わせで実現されることを理解されたい。上述実施形態において、複数のステップまたは方法は、メモリに保存され、且つ適切な命令実行システムによって実行されるソフトウェアまたはファームウェアで実現されることができる。例えば、別の実施形態のようにハードウェアで実現する場合、当技術分野で周知の以下の技術のいずれか一つまたは組み合わせによって実現することができる：データ信号に論理機能を実装する論理ゲートを備えた個別論理回路、適切な組み合わせ論理ゲートを備えた専用集積回路、プログラマブルゲートアレイ（ＰＧＡ）、フィールドプログラマブルゲートアレイ（ＦＰＧＡ）など。

【０１１３】

当業者は、上記実施例を実現する方法を実現するステップのすべてまたは一部は、関連するハードウェアに命令するプログラムによって完了でき、前記プログラムは、コンピュータ可読記憶媒体に保存されることができ、当該プログラムは実行される場合に、方法実施例のステップの一つまたはその組み合わせを含むことを理解できる。

【０１１４】

なお、本出願の各実施例における各機能ユニットは、一つの処理モジュールに統合されてもよく、各ユニットは物理的に別々に存在してもよく、または二つまたはそれ以上のユニットが一つのモジュールに統合されてもよい。上記の統合されたモジュールは、ハードウェアの形態で実現されてもよく、ソフトウェア機能モジュールの形態で実現されてもよい。前記統合されたモジュールは、ソフトウェア機能モジュールの形態で実現され、スタンドアロン製品として販売または使用される場合、コンピュータ可読記憶媒体に保存されることもできる。

【０１１５】

上述の記憶媒体は、読み取り専用メモリ、磁気ディスク、または光ディスクなどである

ことができる。

【０１１６】

本明細書の説明において、「一実施例」、「いくつかの実施例」、「例示」、「具体的な実施例」、または「いくつかの実施例」などの説明は、当該実施例または例示に結合して説明される具体的な特徴、構造、材料または特性が、本出願の少なくとも一つの実施例または例示に含まれることを意味する。本明細書において、上記用語の概略的な表現は、必ずしも同じ実施例または例示を意味するものではない。さらに、説明される具体的な特徴、構造、材料、または特性は、任意の一つまたは複数の実施例または例示において適切な方法で結合されることができる。

【０１１７】

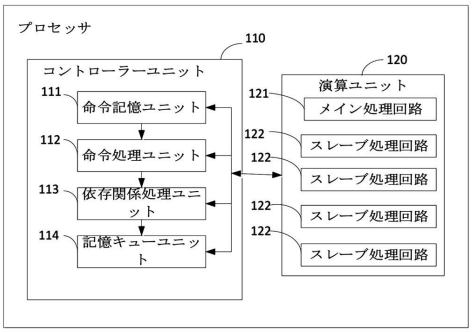
以上で本出願の実施例を示して説明したが、上述の実施例は例示であり、本出願の範囲を限定するものとして解釈されるべきではなく、当業者は、本出願の範囲内で、上記実施例の変化、修正、変更、及び変形を行うことができる。

【図１】

図1

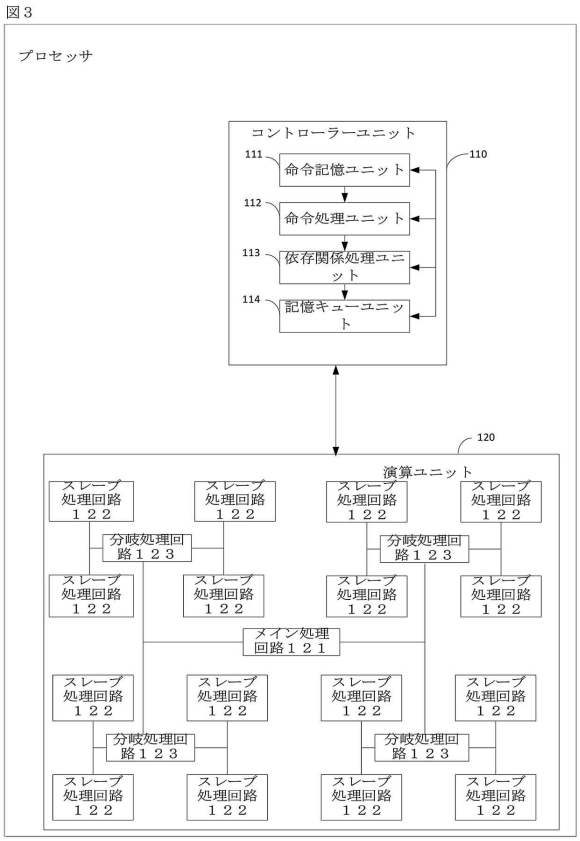
【図２】

図2



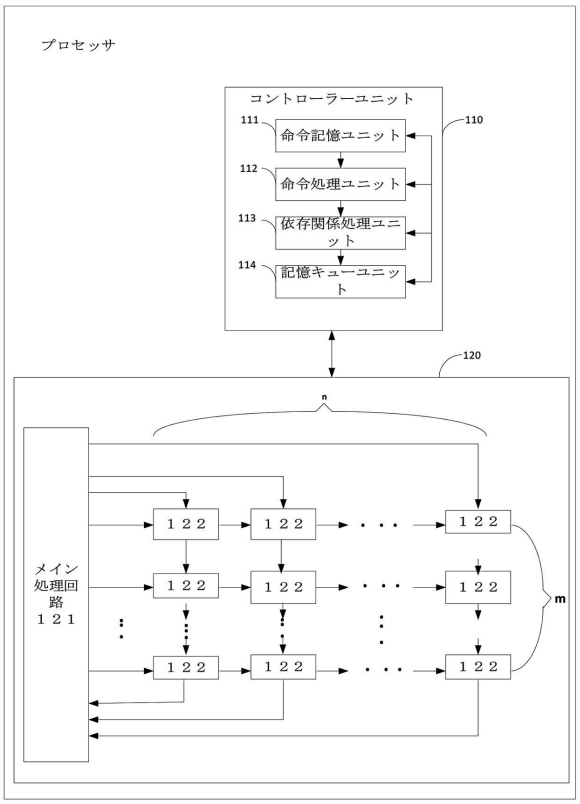
【図３】

図3



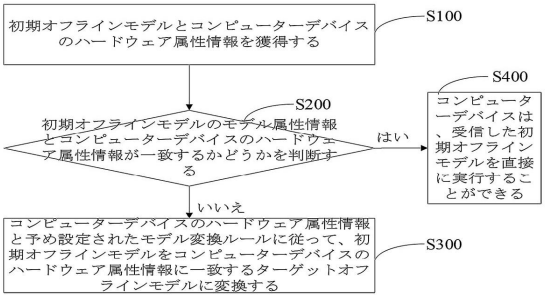
【図４】

図４



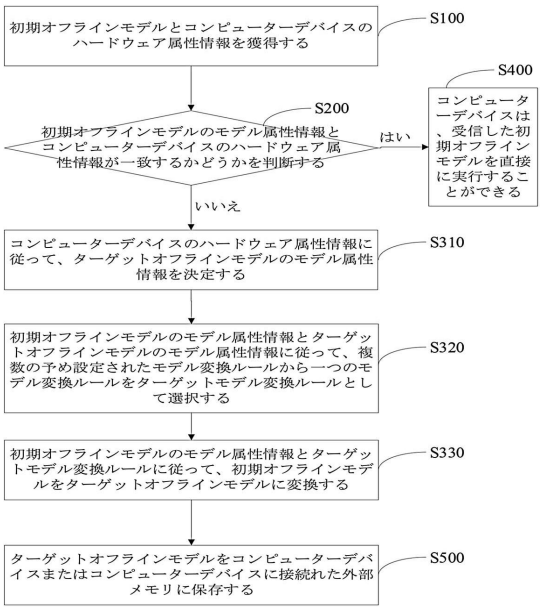
【図５】

図５



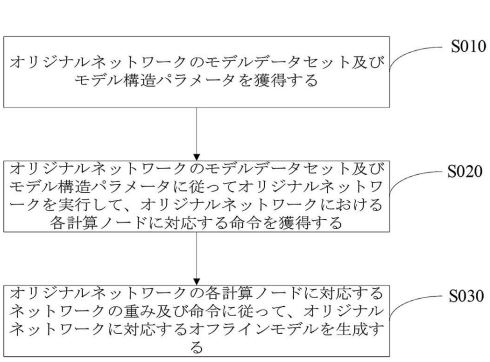
【図６】

図６



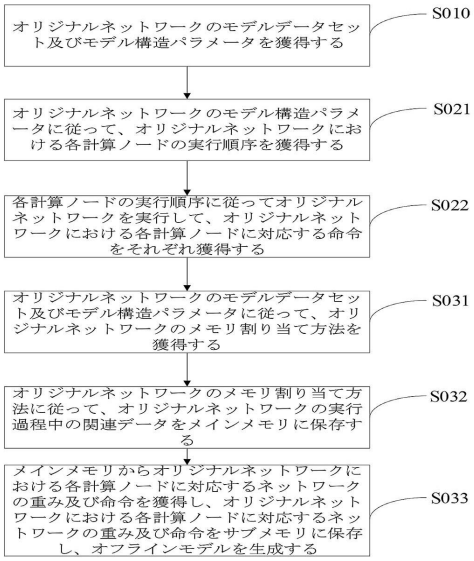
【図７】

図７



【図８】

図８



【図９】

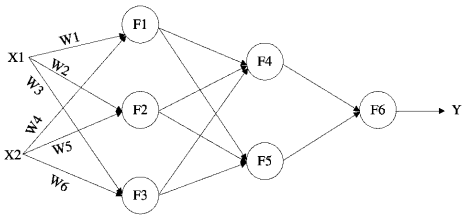
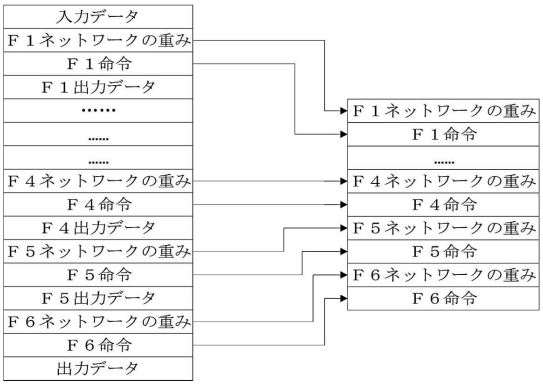


図 9

【図１０】

図１０



フロントページの続き

(74)代理人 100165191
弁理士 河合 章
(74)代理人 100133835
弁理士 河野 努
(72)発明者 リウ シャオリー
中華人民共和国, ベイジン １００１９０, ハイジャン ディストリクト, コーシュエユアン サウス ロード, ナンバー ６, サイエンティフィック リサーチ コンプレックス ビルディング, ルーム ６４４
(72)発明者 リャン チュン
中華人民共和国, ベイジン １００１９０, ハイジャン ディストリクト, コーシュエユアン サウス ロード, ナンバー ６, サイエンティフィック リサーチ コンプレックス ビルディング, ルーム ６４４
(72)発明者 クオ チー
中華人民共和国, ベイジン １００１９０, ハイジャン ディストリクト, コーシュエユアン サウス ロード, ナンバー ６, サイエンティフィック リサーチ コンプレックス ビルディング, ルーム ６４４

審査官 久保 光宏

(56)参考文献 米国特許出願公開第２０１６／０３２８６４４（ＵＳ，Ａ１）
「ソニーも参戦、深層学習ソフト 組み込み向けの開発環境で競う」, 日経エレクトロニクス, 日本, 日経ＢＰ社, ２０１７年 ８月２０日, 2017年9月号（第1183号）, 第６１～６４頁, ISSN: 0385-1680
@Vengineer, 「特集 知っ得 世界のＡＩ技術 第２部第２章 グーグルTensorFlowがいろんなプロセッサに対応できるメカニズム」, インターフェース（Interface）, 日本, ＣＱ出版株式会社, ２０１８年 ２月 １日, 2018年2月号（第44巻, 第2号）, 第６４～７３頁, ISSN: 0387-9569
松尾恒, 「Deep Learning推論の多階層高速化フレームワーク」, 画像ラボ, 日本, 日本工業出版株式会社, ２０１８年 ８月１０日, 2018年8月号（第29巻, 第8号）, 第２９～３７頁, ISSN: 0915-6755
古川新, 「ラズパイで始める人工知能 機械学習で顔認識モデルを作ろう 最終回 TensorFlowモデルの実行方法」, 日経ＢＰパソコンベストムック ラズパイマガジン, 日本, 日経ＢＰ社, ２０１８年 ７月 ９日, 2018年8月号, 第９０～９４頁, ISBN: 978-4-8222-9271-3（文献に記載の発行日：2018年8月22日, (財)ソフトウェア情報センター受入日：2018年7月9日）

(58)調査した分野(Int.Cl., ＤＢ名)
Ｇ０６Ｎ３／００－９９／００
ＣＳＤＢ（日本国特許庁）
ＩＥＥＥＸｐｌｏｒｅ（ＩＥＥＥ）