

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7545461号
(P7545461)

(45)発行日 令和6年9月4日(2024.9.4)

(24)登録日 令和6年8月27日(2024.8.27)

(51)国際特許分類

F I

G 0 6 F 17/16 (2006.01)

G 0 6 F 17/16 Z

G 0 6 N 3/04 (2023.01)

G 0 6 N 3/04

請求項の数 37 (全32頁)

(21)出願番号	特願2022-203468(P2022-203468)	(73)特許権者	514322098
(22)出願日	令和4年12月20日(2022.12.20)		ベijing バイドウ ネットコムサイエ
(65)公開番号	特開2023-36773(P2023-36773A)		ンス テクノロジー カンパニー リミテ
(43)公開日	令和5年3月14日(2023.3.14)		ッド
審査請求日	令和4年12月20日(2022.12.20)		Beijing Baidu Netco
(31)優先権主張番号	202111659099.2		m Science Technolog
(32)優先日	令和3年12月31日(2021.12.31)		y Co., Ltd.
(33)優先権主張国・地域又は機関	中国(CN)		中華人民共和国 ペキン 100085,
			ハイディアン ディストリクト, シャン
			ディ テンス ストリート, 10番, バ
			イドウ キャンパス 2階
			2/F Baidu Campus, N
			o. 10, Shangdi 10th
			Street, Haidian Dis
			trict, Beijing 1000
			最終頁に続く

(54)【発明の名称】 データ処理方法、データ処理装置、電子機器、記憶媒体およびコンピュータプログラム

(57)【特許請求の範囲】

【請求項1】

データ処理装置によって実行されるデータ処理方法であって、

入力データの第1の属性情報を取得し、且つ前記入力データに対応する演算装置の第2の属性情報を取得するステップと、

前記第1の属性情報および前記第2の属性情報に基づいて、候補演算子生成方式からターゲット演算子生成方式を選択するステップと、

前記ターゲット演算子生成方式により、前記第1の属性情報に基づいて、演算子ライブラリから前記入力データに必要な演算子に含まれるサブ演算子を決定して、前記演算子を生成するステップと、

前記演算装置により前記演算子に従って前記入力データの演算を行って、演算結果を取得するステップと、

を含み、

前記入力データが、画像データ、オーディオデータおよびテキストデータのうちの1つであり、前記入力データの第1の属性情報がデータの形状と、データのタイプと、を含み、演算装置の第2の属性情報が、演算装置の型番と、演算装置の加速ライブラリバージョンと、演算装置の環境と、を含み、

前記第1の属性情報および前記第2の属性情報に基づいて、候補演算子生成方式からターゲット演算子生成方式を選択するステップが、

前記第1の属性情報および前記第2の属性情報に基づいて、前記候補演算子生成方式の性

能評価を行って、対応する性能評価結果を取得するステップと、
前記性能評価結果に基づいて、前記候補演算子生成方式から前記ターゲット演算子生成方式を選択するステップと、
を含み、
前記第1の属性情報および前記第2の属性情報に基づいて、前記候補演算子生成方式の性能評価を行って、対応する性能評価結果を取得するステップが、
生成方式に対応する履歴性能評価結果を取得するステップと、
前記第1の属性情報および前記第2の属性情報に基づいて、前記履歴性能評価結果からターゲット性能評価結果を前記性能評価結果として選択するステップと、
を含むデータ処理方法。

10

【請求項2】

前記第1の属性情報および前記第2の属性情報に基づいて、前記履歴性能評価結果からターゲット性能評価結果を前記性能評価結果として選択するステップが、
前記履歴性能評価結果に基づいて、対応する履歴入力データと履歴演算装置とを取得するステップと、
前記履歴性能評価結果における前記履歴入力データの第1の属性情報および前記入力データの第1の属性情報ならびに前記履歴演算装置の第2の属性情報および前記演算装置の第2の属性情報に基づいて、前記履歴性能評価結果からターゲット性能評価結果を前記性能評価結果として選択するステップと、
を含む請求項1に記載のデータ処理方法。

20

【請求項3】

前記履歴性能評価結果における前記履歴入力データの第1の属性情報および前記入力データの第1の属性情報ならびに前記履歴演算装置の第2の属性情報および前記演算装置の第2の属性情報に基づいて、前記履歴性能評価結果からターゲット性能評価結果を前記性能評価結果として選択するステップが、
前記履歴入力データの第1の属性情報と前記入力データの第1の属性情報とを比較するステップと、
前記履歴演算装置の第2の属性情報と前記演算装置の第2の属性情報とを比較するステップと、
前記履歴性能評価結果から、前記入力データの第1の属性情報と一致し且つ前記演算装置の第2の属性情報と一致するターゲット性能評価結果を前記性能評価結果として選択するステップと、
を含む請求項2に記載のデータ処理方法。

30

【請求項4】

前記性能評価結果に基づいて、前記候補演算子生成方式から前記ターゲット演算子生成方式を選択するステップが、
前記性能評価結果に対応する候補演算子生成方式を、前記ターゲット演算子生成方式として決定するステップを含む請求項2に記載のデータ処理方法。

【請求項5】

前記第1の属性情報および前記第2の属性情報に基づいて、前記候補演算子生成方式の性能評価を行って、対応する性能評価結果を取得するステップが、
前記履歴性能評価結果には前記ターゲット性能評価結果が含まれていないことに応答して、前記第1の属性情報、前記第2の属性情報および前記演算子に基づいて、前記候補演算子生成方式の損失情報を取得し、前記損失情報に基づいて前記候補演算子生成方式の性能評価結果を生成するステップを含む請求項3に記載のデータ処理方法。

40

【請求項6】

前記損失情報に基づいて前記候補演算子生成方式の性能評価結果を生成するステップの後に、
前記ターゲット演算子生成方式に対応する性能評価結果を、前記第1の属性情報及び前記第2の属性情報にバインディングして、1つの新しい履歴性能評価結果として記憶する

50

ステップを含む請求項 5 に記載のデータ処理方法。

【請求項 7】

前記性能評価結果には複数の性能指標が含まれ、前記性能評価結果に基づいて、前記候補演算子生成方式からターゲット演算子生成方式を選択するステップが、

前記ターゲット演算子生成方式を選択するためのターゲット性能指標を決定するステップと、

前記ターゲット性能指標および前記性能評価結果に基づいて、前記候補演算子生成方式から前記ターゲット演算子生成方式を選択するステップと、

を含む請求項 5 に記載のデータ処理方法。

【請求項 8】

前記ターゲット性能指標および前記性能評価結果に基づいて、前記候補演算子生成方式から前記ターゲット演算子生成方式を選択するステップが、

性能選好選択ポリシーを選択し、前記性能選好選択ポリシーに基づいてターゲット性能指標を決定するステップと、

前記性能評価結果から、前記ターゲット性能指標にマッチングする指標評価結果を取得するステップと、

前記指標評価結果に基づいて、前記候補演算子生成方式から前記ターゲット演算子生成方式を選択するステップと、

を含む請求項 7 に記載のデータ処理方法。

【請求項 9】

前記ターゲット演算子生成方式に基づいて、演算子ライブラリから前記入力データに必要な演算子に含まれるサブ演算子を決定して、前記演算子を生成するステップが、

前記ターゲット演算子生成方式が演算子組み合わせ実現方式であることに応答して、前記入力データの第 1 の属性情報に基づいて、必要な演算子と、前記必要な演算子に含まれる少なくとも 1 つの第 1 のサブ演算子と、前記第 1 のサブ演算子間の組み合わせ順序とを取得するステップと、

前記組み合わせ順序に基づいて、前記第 1 のサブ演算子を前記演算子に順次組み合わせるステップと、

を含む請求項 1 に記載のデータ処理方法。

【請求項 10】

前記ターゲット演算子生成方式に基づいて、演算子ライブラリから前記入力データに必要な演算子に含まれるサブ演算子を決定して、前記演算子を生成するステップが、

前記ターゲット演算子生成方式が演算子コンパイラ実現方式であることに応答して、前記入力データの第 1 の属性情報に基づいて、必要な演算子と、前記必要な演算子に含まれる少なくとも 1 つの第 2 のサブ演算子と、前記第 2 のサブ演算子のコンパイル順序とを取得するステップと、

すべての前記第 2 のサブ演算子に対してチューニング処理を行って、チューニングされたターゲット第 2 のサブ演算子を取得するステップと、

前記コンパイル順序に基づいて、前記ターゲット第 2 のサブ演算子を順次コンパイルして、前記演算子を生成するステップと、

を含む請求項 1 に記載のデータ処理方法。

【請求項 11】

すべての前記第 2 のサブ演算子に対してチューニング処理を行って、チューニングされたターゲット第 2 のサブ演算子を取得するステップが、

ターゲットチューニングポリシーを取得し、前記ターゲットチューニングポリシーに基づいてすべての前記第 2 のサブ演算子に対して演算子融合を行って、融合サブ演算子を取得するステップと、

前記融合サブ演算子に対して演算子グルーピングおよび演算子スケジューリングを行って、前記ターゲット第 2 のサブ演算子を取得するステップと、

を含む請求項 10 に記載のデータ処理方法。

10

20

30

40

50

【請求項 1 2】

ターゲットチューニングポリシーを取得するステップが、
前記第 1 の属性情報に基づいて、前記入力データのデータ形状を決定するステップと、
前記データ形状に基づいて、対応する前記ターゲットチューニングポリシーを取得する
ステップと、
を含む請求項 1 1 に記載のデータ処理方法。

【請求項 1 3】

前記演算装置により前記演算子に従って前記入力データの演算を行って、演算結果を取
得するステップが、

前記入力データの前記演算子に基づいて、前記入力データに対応するディープニューラ
ルネットワークの演算子セットを生成するステップと、

前記演算子セットを前記演算装置に配布して前記ディープニューラルネットワークを構
築し、前記ディープニューラルネットワークにより前記入力データを演算して、前記演算
結果を取得するステップと、

を含む請求項 1 に記載のデータ処理方法。

【請求項 1 4】

前記演算装置により前記演算子に従って前記入力データの演算を行って、演算結果を取
得するステップが、

演算プロセスを監視し、前記演算プロセスの異常にตอบสนองして、演算を停止し且つ異常ロ
グを生成するステップを含む請求項 1 に記載のデータ処理方法。

【請求項 1 5】

前記演算子の演算プロセスの異常にตอบสนองして、前記演算装置の第 2 の属性情報に基づい
て、新たな第 1 の演算装置をスケジューリングし、前記新たな第 1 の演算装置により前記
演算子に従って前記入力データの演算を行って、演算結果を取得するステップを含む請求
項 1 4 に記載のデータ処理方法。

【請求項 1 6】

前記演算子の演算プロセスの異常にตอบสนองして、前記入力データの第 1 の属性情報に基づ
いて、新たな第 2 の演算装置をスケジューリングし、前記第 2 の演算装置の第 2 の属性情
報を取得し、前記第 1 の属性情報および前記第 2 の属性情報に基づいて候補演算子生成方
式からターゲット演算子生成方式を選択する前記ステップおよびその後のステップに戻る
ステップを含む請求項 1 4 に記載のデータ処理方法。

【請求項 1 7】

入力データの第 1 の属性情報を取得するステップの前に、
データ処理タスクを受信し、前記データ処理タスクから前記入力データを抽出するステ
ップを含む請求項 1 に記載のデータ処理方法。

【請求項 1 8】

入力データの第 1 の属性情報を取得し、且つ前記入力データに対応する演算装置の第 2
の属性情報を取得するように構成される取得モジュールと、

前記第 1 の属性情報および前記第 2 の属性情報に基づいて、候補演算子生成方式からタ
ーゲット演算子生成方式を選択するように構成される選択モジュールと、

前記ターゲット演算子生成方式により、前記第 1 の属性情報に基づいて、演算子ライブ
ラリから入力データに必要な演算子に含まれるサブ演算子を決定して、前記演算子を生成
するように構成される生成モジュールと、

前記演算装置により前記演算子に従って前記入力データの演算を行って、演算結果を取
得するように構成される演算モジュールと、

を備え、

前記入力データが、画像データ、オーディオデータおよびテキストデータのうちの 1 つで
あり、前記入力データの第 1 の属性情報がデータの形状と、データのタイプと、を含み、
演算装置の第 2 の属性情報が、演算装置の型番と、演算装置の加速ライブラリバージョン
と、演算装置の環境と、を含み、

10

20

30

40

50

前記選択モジュールが、
前記第1の属性情報および前記第2の属性情報に基づいて、前記候補演算子生成方式の性能評価を行って、対応する性能評価結果を取得して、
前記性能評価結果に基づいて、前記候補演算子生成方式から前記ターゲット演算子生成方式を選択するように構成され、
前記選択モジュールが、
前記演算子の前記候補演算子生成方式に対応する履歴性能評価結果を取得し、
前記第1の属性情報および前記第2の属性情報に基づいて、前記履歴性能評価結果からターゲット性能評価結果を前記性能評価結果として選択するように構成されるデータ処理装置。

10

【請求項19】

前記選択モジュールが、
前記履歴性能評価結果に基づいて、対応する履歴入力データと履歴演算装置とを取得し、
前記履歴性能評価結果における前記履歴入力データの第1の属性情報および前記入力データの第1の属性情報ならびに前記履歴演算装置の第2の属性情報および前記演算装置の第2の属性情報に基づいて、前記履歴性能評価結果からターゲット性能評価結果を前記性能評価結果として選択するように構成される請求項18に記載のデータ処理装置。

【請求項20】

前記選択モジュールが、
前記履歴入力データの第1の属性情報と前記入力データの第1の属性情報とを比較し、
前記履歴演算装置の第2の属性情報と前記演算装置の第2の属性情報とを比較し、
前記履歴性能評価結果から、前記入力データの第1の属性情報と一致し且つ前記演算装置の第2の属性情報と一致するターゲット性能評価結果を前記性能評価結果として選択するように構成される請求項19に記載のデータ処理装置。

20

【請求項21】

前記選択モジュールが、
前記性能評価結果に対応する候補演算子生成方式を前記ターゲット演算子生成方式として決定するように構成される請求項19または20に記載のデータ処理装置。

【請求項22】

前記選択モジュールが、
前記履歴性能評価結果にはターゲット性能評価結果が含まれていないことに応答して、
前記第1の属性情報、前記第2の属性情報および前記演算子に基づいて、前記候補演算子生成方式の損失情報を取得し、前記損失情報に基づいて前記候補演算子生成方式の性能評価結果を生成するように構成される請求項20に記載のデータ処理装置。

30

【請求項23】

前記選択モジュールが、
前記ターゲット演算子生成方式に対応する性能評価結果を、前記第1の属性情報および前記第2の属性情報にバインディングして、1つの新しい履歴性能評価結果として記憶するように構成される請求項22に記載のデータ処理装置。

【請求項24】

前記選択モジュールが、
前記ターゲット演算子生成方式を選択するように構成されるターゲット性能指標を決定し、
前記ターゲット性能指標と前記性能評価結果とに基づいて、前記候補演算子生成方式から前記ターゲット演算子生成方式を選択するように構成される請求項22に記載のデータ処理装置。

40

【請求項25】

前記選択モジュールが、
性能選好選択ポリシーを取得し、前記性能選好選択ポリシーに基づいて、ターゲット性能指標を決定し、

50

前記性能評価結果から、前記ターゲット性能指標にマッチングする指標評価結果を選択し、

前記指標評価結果に基づいて、前記候補演算子生成方式から前記ターゲット演算子生成方式を選択するように構成される請求項 2 4 に記載のデータ処理装置。

【請求項 2 6】

前記生成モジュールが、

前記ターゲット演算子生成方式が演算子組み合わせ実現方式であることに応答して、前記入力データの第 1 の属性情報に基づいて、必要な演算子と、前記必要な演算子に含まれる少なくとも 1 つの第 1 のサブ演算子と、前記第 1 のサブ演算子間の組み合わせ順序とを取得し、

前記組み合わせ順序に基づいて、前記第 1 のサブ演算子を前記演算子に順次組み合わせるように構成される請求項 1 8 に記載のデータ処理装置。

【請求項 2 7】

前記生成モジュールが、

前記ターゲット演算子生成方式が演算子コンパイラ実現方式であることに応答して、前記入力データの第 1 の属性情報に基づいて、必要な演算子と、前記必要な演算子に含まれる少なくとも 1 つの第 2 のサブ演算子と、前記第 2 のサブ演算子のコンパイル順序とを取得し、

すべての前記第 2 のサブ演算子に対してチューニング処理を行って、チューニングされたターゲット第 2 のサブ演算子を取得し、

前記コンパイル順序に基づいて、前記ターゲット第 2 のサブ演算子を順次コンパイルして、前記演算子を生成するように構成される請求項 1 8 に記載のデータ処理装置。

【請求項 2 8】

前記生成モジュールが、

ターゲットチューニングポリシーを取得し、前記ターゲットチューニングポリシーに基づいてすべての前記第 2 のサブ演算子に対して演算子融合を行って、融合サブ演算子を取得し、

前記融合サブ演算子に対して演算子グルーピングおよび演算子スケジューリングを行って、前記ターゲット第 2 のサブ演算子を取得するように構成される請求項 2 7 に記載のデータ処理装置。

【請求項 2 9】

前記生成モジュールが、

前記第 1 の属性情報に基づいて、前記入力データのデータ形状を決定し、

前記データ形状に基づいて、対応する前記ターゲットチューニングポリシーを取得するように構成される請求項 2 8 に記載のデータ処理装置。

【請求項 3 0】

前記演算モジュールが、

前記入力データの前記演算子に基づいて、前記入力データに対応するディープニューラルネットワークの演算子セットを生成し、

前記演算子セットを前記演算装置に配布して前記ディープニューラルネットワークを構築し、前記ディープニューラルネットワークにより前記入力データを演算して、前記演算結果を取得するように構成される請求項 1 8 に記載のデータ処理装置。

【請求項 3 1】

前記演算モジュールが、

演算プロセスを監視し、前記演算プロセスの異常に応答して、演算を停止し且つ異常ログを生成するように構成される請求項 1 8 に記載のデータ処理装置。

【請求項 3 2】

前記演算モジュールが、

前記演算装置の第 2 の属性情報に基づいて、新たな第 1 の演算装置をスケジューリングし、前記新たな第 1 の演算装置により前記演算子に従って前記入力データの演算を行って

10

20

30

40

50

、演算結果を取得するように構成される請求項 3 1 に記載のデータ処理装置。

【請求項 3 3】

前記演算モジュールが、

前記入力データの第 1 の属性情報に基づいて、新たな第 2 の演算装置をスケジューリングし、前記第 2 の演算装置の第 2 の属性情報を取得し、前記第 1 の属性情報および前記第 2 の属性情報に基づいて候補演算子生成方式からターゲット演算子生成方式を選択するステップおよびその後のステップに戻るように構成される請求項 3 1 に記載のデータ処理装置。

【請求項 3 4】

前記取得モジュールが、

データ処理タスクを受信し、前記データ処理タスクから前記入力データを抽出するように構成される請求項 1 8 に記載のデータ処理装置。

【請求項 3 5】

少なくとも 1 つのプロセッサと、

該少なくとも 1 つのプロセッサと通信可能に接続されるメモリと、
を備え、

前記メモリには、前記少なくとも 1 つのプロセッサによって実行可能な命令が記憶されており、前記命令が前記少なくとも 1 つのプロセッサによって実行される場合、前記少なくとも 1 つのプロセッサが請求項 1 から 1 7 のいずれか一項に記載のデータ処理方法を実行できる電子機器。

【請求項 3 6】

コンピュータに請求項 1 から 1 7 のいずれか一項に記載のデータ処理方法を実行させるためのコンピュータ命令をした非一時的コンピュータ読み取り可能な記憶媒体。

【請求項 3 7】

プロセッサによって実行される場合、請求項 1 から 1 7 のいずれか一項に記載のデータ処理方法のステップを実現するコンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本開示は、データ処理技術分野に関し、具体的には、ビッグデータ、クラウドコンピューティング、ディープラーニングなどの人工知能技術分野に関し、特にデータ処理方法、データ処理装置、電子機器、記憶媒体およびコンピュータプログラムに関する。

【背景技術】

【0002】

ディープラーニングフレームワークはディープラーニング技術の研究開発に使用される基本的なソフトウェアツールであり、上流に対してアプリケーションに対応し、下流に対してハードウェアデバイスに対応し、ディープラーニングタスクの研究開発効率を向上させることができる。ディープラーニングフレームはハードウェアの相違を隠し、アプリケーションの開発に統一されたインターフェースを提供する。

【0003】

しかしながら、ディープラーニングフレームワークに含まれる演算子の数は非常に多く、関連技術では、コストが低く且つハードウェア性能を十分に向上させることができる演算子生成方式は未だ存在し提供ない。そのため、演算結果を正確に取得することを確保するとともにハードウェア性能を向上させてコストを低減することを如何に実現するかは、早急に解決すべき問題となっている。

【発明の概要】

【0004】

本開示は、データ処理方法、データ処理装置、電子機器、記憶媒体およびコンピュータプログラムを提供する。

【0005】

本開示の第1の態様では、入力データの第1の属性情報を取得し、且つ前記入力データに対応する演算装置の第2の属性情報を取得するステップと、前記第1の属性情報および前記第2の属性情報に基づいて、候補演算子生成方式からターゲット演算子生成方式を選択するステップと、前記ターゲット演算子生成方式に基づいて、演算子ライブラリから前記入力データに必要な演算子に含まれるサブ演算子を決定して、前記演算子を生成するステップと、前記演算装置により前記演算子に従って前記入力データの演算を行って、演算結果を取得するステップとを含むデータ処理方法が提供される。

【0006】

本開示の第2の態様では、入力データの第1の属性情報を取得し、且つ前記入力データ演算子に対応する演算装置の第2の属性情報を取得するように構成される取得モジュールと、前記第1の属性情報および前記第2の属性情報に基づいて、候補演算子生成方式からターゲット演算子生成方式を選択するように構成される選択モジュールと、前記ターゲット演算子生成方式に基づいて、演算子ライブラリから前記入力データに必要な演算子に含まれるサブ演算子を決定して、前記演算子を生成するように構成される生成モジュールと、前記演算装置により前記演算子に従って前記入力データの演算を行って、演算結果を取得するように構成される演算モジュールとを備えるデータ処理装置が提供される。

【0007】

本開示の第3の態様では、少なくとも1つのプロセッサと、該少なくとも1つのプロセッサと通信可能に接続されるメモリと、を備え、前記メモリには、前記少なくとも1つのプロセッサによって実行可能な命令が記憶されており、前記命令が前記少なくとも1つのプロセッサによって実行される場合、前記少なくとも1つのプロセッサが本開示の第1の態様に記載のデータ処理方法を実行できる電子機器が提供される。

【0008】

本開示の第4の態様では、コンピュータに本開示の第1の態様に記載のデータ処理方法を実行させるためのコンピュータ命令を記憶した非一時的コンピュータ読み取り可能な記憶媒体が提供される。

【0009】

本開示の第5の態様では、プロセッサによって実行される場合、本開示の第1の態様に記載のデータ処理方法を実現するコンピュータプログラムが提供される。

【0010】

なお、この概要部分で説明された内容は本開示の実施例の肝心または重要な特徴を特定しようとするわけではなく、本開示の範囲を限定するわけでもない。本開示の他の特徴は以下の明細書により理解しやすくなる。

【図面の簡単な説明】

【0011】

図面は本解決案をよりよく理解するためのものであり、本開示を限定するものではない。

【図1】 本開示の第1の実施例に係るデータ処理方法の概略フローチャートである。

【図2】 本開示の第2の実施例に係るデータ処理方法の概略フローチャートである。

【図3】 本開示の第3の実施例に係るデータ処理方法の概略フローチャートである。

【図4】 本開示の第4の実施例に係るデータ処理方法の概略フローチャートである。

【図5】 本開示の第5の実施例に係るデータ処理方法の概略フローチャートである。

【図6】 本開示の第6の実施例に係るデータ処理方法の概略フローチャートである。

【図7】 本開示の第7の実施例に係るデータ処理方法の概略フローチャートである。

【図8】 本開示の第8の実施例に係るデータ処理方法の概略フローチャートである。

【図9】 本開示の第9の実施例に係るデータ処理方法の概略フローチャートである。

【図10】 本開示に係る1つのデータ処理フレームの概略図である。

【図11】 本開示に係るデータ処理方法の適用概略図の一つである。

【図12】 本開示に係るデータ処理方法の適用概略図の一つである。

【図13】 本開示に係るデータ処理方法の適用概略図の一つである。

【図14】 本開示の実施例に係るデータ処理方法を実現するためのデータ処理装置のプロ

10

20

30

40

50

ック図である。

【図 15】本開示の実施例のデータ処理方法を実現するための電子機器のブロック図である。

【発明を実施するための形態】

【0012】

以下、図面を参照しながら本開示の例示的な実施例を説明し、理解を助けるために、その中には本開示の実施例の様々な詳細が含まれており、それらを例示的なものとして見なすべきである。そのため、当業者であれば、本開示の範囲と趣旨から逸脱しない限り、ここで説明される実施例に対して様々な変更と修正を行うことができる。同様に、明確と簡潔のために、以下の説明では周知機能および構造に関する説明を省略している。

10

【0013】

データ処理 (Data Processing) は、データの収集、記憶、検索、加工、変化および伝送などの処理を含み、乱雑で且つ理解し難い大量のデータから、特定のユーザにとって価値と意義のあるデータを抽出し且つ導出することを目的としている。

【0014】

ビッグデータ (Big Data) は、膨大量資料とも呼ばれ、関わる資料の量の規模が大きすぎて、主流ソフトウェアツールを通じて合理的な時間内に抽出、管理、処理して、企業の経営及び意思決定に役立つ情報に整理できない資料を指す。

【0015】

クラウドコンピューティング (Cloud Computing) は分散コンピューティングの1つであり、ネットワーク「クラウド」を介して大量のデータ計算処理プログラムを無数の小さいプログラムに分解し、その後、複数のサーバで構成されるシステムを介してこれらの小さいプログラムを処理分析し、結果を取得してユーザに返すことを指す。

20

【0016】

人工知能 (Artificial Intelligence、AIと略称する) は、人間の知能をシミュレーションし且つ拡張する理論、方法、技術およびアプリケーションシステムを研究開発する技術科学である。現在では、AI技術は自動化程度が高く、精度が高く、コストが低いという利点があり、幅広く適用されている。

【0017】

ディープラーニング (Deep Learning、DLと略称する) は、機械学習 (Machine Learning、MLと略称する) 分野における新しい研究方向であり、サンプルデータの固有法則と表現階層を学習し、これらの学習プロセスで得られた情報は文字、画像と音声などのデータの解釈に大いに役立つ。その最終の目的は、ロボットが人間のように分析学習能力を備え、文字、画像と音声などのデータを認識できるようにすることである。具体的な研究内容として、主に畳み込み演算に基づくニューロンネットワークシステム、即ち畳み込みニューロンネットワークと、多層ニューロンに基づく自己符号化ニューロンネットワークとを含み、多層自己符号化ニューロンネットワークの方式で事前訓練を行って、さらに識別情報に合わせてニューロンネットワーク重みの深層信頼ネットワークを最適化する。ディープラーニングは検索技術、データマイニング、機械学習、機械翻訳、自然言語処理、マルチメディア学習、音声、推薦と個性化技術、および他の関連分野で多くの成果を出している。

30

【0018】

以下、図面を参照して本開示の実施例に係るデータ処理方法を説明する。

【0019】

図 1 は、本開示の第 1 の実施例に係るデータ処理方法の概略フローチャートである。ここで、本実施例のデータ処理方法の実行主体はデータ処理装置であり、データ処理装置は具体的にハードウェアデバイス、またはハードウェアデバイスにおけるソフトウェアなどであってもよい。ハードウェアデバイスは例えば端末装置、サーバ等であってもよい。

40

【0020】

図 1 に示すように、本開示の実施例に係るデータ処理方法は、具体的には、以下のステ

50

ップ S 1 0 1 ~ S 1 0 4 を含むことができる。

【 0 0 2 1 】

S 1 0 1 において、入力データの第 1 の属性情報を取得し、且つ入力データに対応する演算装置の第 2 の属性情報を取得する。

【 0 0 2 2 】

演算子は 1 つの関数空間から関数空間へのマッピングを指す。

【 0 0 2 3 】

入力データは、演算を行う必要があるデータを指し、主にテンソル (T e n s o r) を含む。テンソルは、データのタイプ、データの形状、データの保存位置などを含むことができる。

【 0 0 2 4 】

例えば、データのタイプは f l o a t 3 2 、 f l o a t 6 4 、 i n t 3 2 、 i n t 6 4 などのタイプのデータを含むことができ、データの形状はビッグデータ、スモールデータ、ワイドデータ、ディープデータなどを含むことができ、データの保存位置はグラフィック処理装置 (G r a p h i c s P r o c e s s i n g U n i t 、 GPU と略称する) のメモリ、ディープコンピューティングユニット (D e e p C o m p u t i n g U n i t 、 DCU と略称する) のグラフィックメモリおよび特定用途向け集積回路 (A p p l i c a t i o n S p e c i f i c I n t e g r a t e d C i r c u i t 、 ASIC と略称する) メモリ内を含むことができる。

【 0 0 2 5 】

さらに、入力データのソースは、画像、オーディオ、ドキュメントなどであってもよい。入力データに対応する演算子は 1 つまたは複数であってもよい。なお、入力データに対応する各演算子のそれぞれは本開示の実施例によって提供される方式で生成することができる。

【 0 0 2 6 】

ここで、演算装置は、入力データを演算する装置を指す。例えば、演算装置は、GPU、CPU、専用人工知能チップ、例えば、ディープコンピューティングユニット DCU などを備えることができる。

【 0 0 2 7 】

なお、関連技術では、演算子生成方式のスケジューリングルールがシンプルで、ハードウェアの性能を十分に生かすことができず、一方で、演算子生成方式はスケジューリング時に入力データの形状とタイプを考慮しておらず、異なる形状とタイプは同じ演算子生成方式を用い、しかしながら、異なる演算子生成方式は、異なる入力データの形状とタイプが与えられ、パフォーマンスが異なる。他方で、演算子生成方式はスケジューリング時に演算装置の具体的な型番を区分しておらず、例えば、V 1 0 0 、 A 1 0 0 および 3 0 9 0 などの異なるグラフィックカード型番は、異なる演算子生成方式が異なる型番のグラフィックカードで実現される性能が異なる可能性があり、演算装置加速ライブラリのバージョンを考慮しておらず、例えば、同じ演算子生成方式は、加速ライブラリ (c u d n n) のバージョンが異なると、性能が異なる可能性がある。また、演算子生成方式はスケジューリング時に、演算装置の演算環境を考慮しておらず、例えば、同じ GPU のグラフィックカードは、異なるメインボードと CPU における性能が異なる可能性がある。

【 0 0 2 8 】

これにより、本開示では、演算子生成方式スケジューリングルールの情報、即ち入力データの第 1 の属性情報と演算装置の第 2 の属性情報とを取得して、演算子生成方式のスケジューリングを行うことにより、ハードウェアの性能を十分に生かすことができる。

【 0 0 2 9 】

なお、データの第 1 の属性情報はデータの形状、データのタイプなどを含むことができる。

【 0 0 3 0 】

データの形状は、ビク、スモール、ワイド、ディープなどを含むことができ、即ちビ

10

20

30

40

50

ッグデータ、スモールデータ、ワイドデータ、ディープデータなどを含むことができる。

【0031】

データのタイプは、浮動小数点型データ、固定小数点型データなどを含むことができ、浮動小数点型データの大きさは32ビット、64ビットであってもよく、固定小数点型データの大きさは32ビット、64ビットであってもよい。

【0032】

なお、演算装置の第2の属性情報は、演算装置の型番、演算装置の加速ライブラリバージョン、演算装置の環境などを含むことができる。

【0033】

S102において、第1の属性情報と第2の属性情報とに基づいて、候補演算子生成方式からターゲット演算子生成方式を選択する。

10

【0034】

なお、候補演算子生成方式は、2種類以上の演算子生成方式を含むことができ、例えば、演算子組み合わせ実現方式、演算子コンパイラ実現方式を含むことができる。これは例に過ぎず、他の演算子実現も候補演算子実現とすることができる。ここで、演算子組み合わせ実現方式については、小演算子組み合わせを用いて算出することができるため、シンプルで柔軟である。演算子コンパイラ実現方式については、演算子の数を増やさずに済むため、実現性能が高い。

【0035】

なお、本開示では、候補演算子生成方式が予め設定されており、候補演算子生成方式は演算子組み合わせ実現方式または演算子コンパイラ実現方式であってもよい。

20

【0036】

演算子組み合わせ実現方式とは、他のより小さな演算子を用いて、複雑な演算子を組み合わせることを指す。

【0037】

演算子コンパイラ実現方式とは同様に、他のより小さな演算子を用いて、複雑な演算子を組み合わせることを指すが、小さな演算子をチューニングすることになっている。

【0038】

本開示では、第1の属性情報と第2の属性情報とに基づいて、ユーザのニーズ情報を取得することができ、さらには、候補演算子生成方式からターゲット演算子生成方式を選択することにより、演算子組み合わせ実現方式と演算子コンパイラ実現方式とが1つのシステムで共存することを実現し、即ち演算子ライブラリ内部には演算子組み合わせ実現方式と演算子コンパイラ実現方式が同時に存在し、優勢が互いに補い合うことになる。

30

【0039】

S103において、ターゲット演算子生成方式に基づいて、演算子ライブラリから入力データに必要な演算子に含まれるサブ演算子を決定して、演算子を生成する。

【0040】

演算子ライブラリは、すべての演算子のセットを指す。演算子のセット内の演算子は複雑な演算子を含んでもよいし、サブ演算子を含んでもよく、サブ演算子は複雑な演算子を形成するために使用することができる。演算子のセット内は正規化指数（softmax）などを含むことができ、サブ演算子は指数（exponential、expと略称する）、合計（summation、sumと略称する）、割り算（divide）などを含むことができる。

40

【0041】

本開示の実施例では、ターゲット演算子生成方式を取得した後、ターゲット演算子生成方式に基づいて、演算子ライブラリから入力データに必要な演算子（複雑な演算子）を決定することができ、これにより、演算子に基づいてサブ演算子を取得し、サブ演算子に基づいて演算子を生成する。

【0042】

なお、演算子組み合わせ実現方式に基づいて、サブ演算子を組み合わせ、演算子を生成

50

することができる。

【0043】

なお、演算子コンパイラ実現方式に対して、演算子を分解し且つ自動的にチューニングして実行コードを生成することで、チューニングされたサブ演算子を取得して演算子を生成することができる。

【0044】

例えば、演算子を分解し、且つ予め設定されたチューニングルールまたはチューニングテンプレートなどに基づいてチューニングして実行コードを生成することで、演算子を生成することができる。

【0045】

なお、演算子の入力データの第1の属性情報、即ち入力データのタイプ、データの形状および／または演算ニーズに基づいて、入力データに対応する演算子に含まれ得るサブ演算子を決定することができる。ここで、サブ演算子は演算子ライブラリにおける演算子である。1つの入力データは1つまたは複数の演算子に対応することができる。

【0046】

S104において、演算装置により演算子に従って入力データの演算を行って、演算結果を取得する。

【0047】

本開示の実施例では、演算子を取得した後、演算装置にロードまたはリリースすることができ、そして演算装置は演算子に従って入力データの演算を行って演算結果を取得することができる。なお、演算装置は演算子の媒体として、入力データを演算子に入力して関連関数演算を行って、1つの演算結果を取得することができる。

【0048】

以上により、本開示の実施例のデータ処理方法は、入力データの第1の属性情報を取得し、且つ入力データに対応する演算装置の第2の属性情報を取得し、第1の属性情報と第2の属性情報とに基づいて、候補演算子生成方式からターゲット演算子生成方式を選択し、その後、ターゲット演算子生成方式に基づいて、演算子ライブラリから入力データに必要な演算子に含まれるサブ演算子を決定して、演算子を生成し、最後に、演算装置により演算子に従って入力データの演算を行って、演算結果を取得することができる。本開示の実施例では、データの第1の属性情報と演算装置の第2の属性情報とに基づいて、演算子生成方式の自動的なスケジューリングを行うことができ、即ち、入力データと演算装置との異なる組み合わせに対して、最適な演算子生成方式にスケジューリングすることができ、演算装置の演算性能および入力データの演算効率を向上させることができ、また、異なる演算子生成方式がそれぞれの優勢を有するため、上記自動スケジューリングポリシーにより、各演算子生成方式を十分に利用して、入力データに対応する最適な演算子を生成することができ、演算装置のハードウェア性能を十分に生かすことができる。

【0049】

図2は、本開示の第2の実施例に係るデータ処理方法概略フローチャートである。図2に示すように、上記実施例のもとに、本開示の実施例に係るデータ処理方法は、具体的には以下のステップS201～S205を含むことができる。

【0050】

S201において、入力データの第1の属性情報を取得し、且つ入力データに対応する演算装置の第2の属性情報を取得する。

【0051】

当該ステップS201に対して、本開示の各実施例におけるいずれか1つの実現形態で実現することができ、ここで詳しい説明を省略する。

【0052】

S202において、第1の属性情報と第2の属性情報とに基づいて、候補演算子生成方式の性能評価を行って、対応する性能評価結果を取得する。

【0053】

選択可能に、第1の属性情報と第2の属性情報とを性能評価モデルに入力して、対応する評価結果を出力することができる。

【0054】

なお、本開示では、第1の属性情報と第2の属性情報とに基づいて、候補演算子生成方式の性能評価を行う具体的な方式が限定られず、実際の状況に応じて選択することができる。

【0055】

S203において、性能評価結果に基づいて、候補演算子生成方式からターゲット演算子生成方式を選択する。

【0056】

なお、候補演算子生成方式と性能評価結果とは一対一で対応しており、性能評価結果は複数の性能指標を含むことができ、例えば、演算にかかる時間が短く、演算時にリソース消費が少ない等がある。そのため、性能評価結果を取得した後、複数の性能指標のうちの1つまたは複数に基づいて、候補演算子生成方式から1つのターゲット演算子生成方式を選択することができる。例えば、かかる時間が最も短い演算子生成方式を選択しようとする場合、各かかる時間指標を比較して、その中からかかる時間が最も短い性能評価結果を選択することでき、当該かかる時間が最も短い性能評価結果に対応する候補演算子生成方式をターゲット演算子生成方式として決定する。

【0057】

S204において、ターゲット演算子生成方式に基づいて、演算子ライブラリから入力データに必要な演算子に含まれるサブ演算子を決定して、演算子を生成する。

【0058】

S205において、演算装置により演算子に従って入力データの演算を行って、演算結果を取得する。

【0059】

上記ステップS204～S205に対して、本開示の各実施例におけるいずれか1つの実現形態で実現することができ、ここで詳しい説明を省略する。

【0060】

以上により、本開示の実施例に係るデータ処理方法は、第1の属性情報と第2の属性情報とに基づいて、候補演算子生成方式の性能評価を行って、対応する性能評価結果を取得し、性能評価結果に基づいて、候補演算子生成方式からターゲット演算子生成方式を選択することができる。したがって、異なる性能評価結果に対して、それに対応するターゲット演算子生成方式を選択することができるため、ハードウェアインターフェース時の演算子開発の作業量を低減することができ、同時に演算子生成方式を自動的にスケジューリングすることができるため、演算子演算結果の取得効率をさらに向上させることができる。

【0061】

1つの可能な実現形態として、図3に示すように、上記実施例のもとに、第1の属性情報と第2の属性情報とに基づいて、候補演算子生成方式の性能評価を行って、対応する性能評価結果を取得するステップは、以下のステップS301～S302を含むことができる。

【0062】

S301において、演算子の候補演算子生成方式に対応する履歴性能評価結果を取得する。

【0063】

また、候補演算子生成方式に対応する履歴性能評価結果は、演算子組み合わせ実現方式に対応する履歴性能評価結果、または演算子コンパイラ実現方式に対応する履歴性能評価結果である。

【0064】

なお、演算子、入力データおよび固定のソフトウェアハードウェア環境を取得した後、演算子ライブラリバックエンドの演算子組み合わせ実現方式または演算子コンパイラ実現方

10

20

30

40

50

式をそれぞれ実行して、候補演算子生成方式に対応する履歴性能評価結果をそれぞれ取得することができる。

【0065】

履歴性能評価結果は、候補実現方式に対応する複数の指標および指標評価結果を含む。

【0066】

例えば、履歴性能評価結果に含まれる指標は時間指標、消費電力指標およびグラフィックメモリ指標などであってもよい。履歴性能評価結果に含まれる指標評価結果は、かかる時間の長さ、消費電量の大きさ、およびグラフィックメモリ使用量の大きさなどであってもよい。

【0067】

なお、本開示では、演算子の候補演算子生成方式に対応する履歴性能評価結果を取得する具体的な方式が限定されず、実際の状況に応じて選択することができる。

【0068】

選択可能に、キャッシュメモリまたはサーバから履歴性能評価結果を取得することができる。

【0069】

選択可能に、理論推定または性能評価モデルによる予測により、履歴性能評価結果を取得することができる。

【0070】

S302において、第1の属性情報と第2の属性情報とに基づいて、履歴性能評価結果からターゲット性能評価結果を性能評価結果として選択する。

【0071】

また、履歴性能評価結果が第1の属性情報および第2の属性情報に一対一に対応しているため、第1の属性情報と第2の属性情報とに基づいて、履歴性能評価結果からターゲット性能評価結果を性能評価結果として選択することができる。

【0072】

なお、本開示では、履歴性能評価結果からターゲット性能評価結果を性能評価結果として選択する具体的な方式が限定されず、実際の状況に応じて選択することができる。

【0073】

選択可能に、ユーザの実際のニーズに応じて、履歴性能評価結果からターゲット性能評価結果を性能評価結果として選択することができる。例えば、ユーザのニーズが短いかかる時間である場合、履歴性能評価結果からかかる時間が最も短い性能評価結果を性能評価結果として選択することができる。又は、ユーザのニーズが小さい消費電力である場合、履歴性能評価結果から消費電力が最も小さい性能評価結果を性能評価結果として選択することができる。

【0074】

以上により、本開示に係るデータ処理方法は、演算子の候補演算子生成方式に対応する履歴性能評価結果を取得し、第1の属性情報と第2の属性情報とに基づいて、履歴性能評価結果からターゲット性能評価結果を性能評価結果として選択することができる。これにより、第1の属性情報と第2の属性情報とに基づいて性能評価結果を選択することができ、性能評価結果の精度および信頼性を向上させることができる。

【0075】

さらに、履歴性能評価結果を取得した後、履歴性能評価結果からターゲット性能評価結果を性能評価結果として選択して、性能評価結果を取得することができる。

【0076】

1つの可能な実現形態として、図4に示すように、上記実施例のもとに、具体的にh以下のステップS401～S402を含むことができる。

【0077】

S401において、履歴性能評価結果に基づいて、対応する履歴入力データと履歴演算装置とを取得する。

10

20

30

40

50

【0078】

なお、履歴性能評価結果には、履歴入力データおよび履歴演算装置の情報が記録されており、本開示の実施例では、履歴性能評価結果内の記録情報に基づいて、当該履歴性能評価結果に対応する履歴入力データと履歴演算装置とを決定することができる。

【0079】

S402において、履歴性能評価結果における履歴入力データの第1の属性情報および入力データの第1の属性情報と、履歴演算装置の第2の属性情報および演算装置の第2の属性情報とに基づいて、履歴性能評価結果からターゲット性能評価結果を性能評価結果として選択する。

【0080】

本開示の実施例では、性能評価結果を取得するための作業量を低減するために、履歴性能評価結果に含まれる履歴入力データの第1の属性情報および入力データの第1の属性情報と、履歴演算装置の第2の属性情報および演算装置の第2の属性情報とに基づいて、履歴性能評価結果からターゲット性能評価結果を性能評価結果として選択することができる。

【0081】

1つの可能な実現形態として、履歴入力データの第1の属性情報と入力データの第1の属性情報とを比較することにより、現在の入力データ属性情報に一致する履歴入力データがあるか否かを判断する。さらに、履歴演算装置の第2の属性情報と演算装置の第2の属性情報とを比較することにより、現在の演算装置の属性情報と一致する履歴演算装置があるか否かを判断する。

【0082】

本開示の実施例では、履歴性能評価結果から、入力データの第1の属性情報と一致し且つ演算装置の第2の属性情報と一致するターゲット性能評価結果を性能評価結果として選択する必要がある。

【0083】

なお、履歴入力データの第1の属性情報が現在の入力データの第1の属性情報と一致し、且つ履歴演算装置の第2の属性情報が現在の演算装置の第2の属性情報と一致する場合は、類似する演算装置を用いて類似する入力データを処理したことがあることを表し、当該履歴性能評価結果を直接性能評価結果とすることができる。

【0084】

さらに、ターゲット性能評価結果を決定した後、当該ターゲット性能評価結果に対応する候補演算子生成方式を決定することができ、当該ターゲット性能評価結果に対応する候補演算子生成方式を、ターゲット演算子生成方式として決定する。

【0085】

たとえば、演算子Aの履歴入力データ第1の属性情報がa1で、履歴演算装置第2の属性情報がa2で、履歴評価結果がa3で、演算子Bの第1の属性情報がa1と一致し、第2の属性情報がa2と一致する場合、演算子Bの性能評価結果はa3である。例えば、性能評価結果a3に対応する候補演算子生成方式が演算子組み合わせ実現方式である場合、a3で採用した演算子組み合わせ実現方式を直接ターゲット演算子生成方式として決定することができる。選択可能に、履歴性能評価結果にはターゲット性能評価結果が含まれていないことに応答して、即ち、現在の入力データの第1の属性情報と一致する履歴入力データが存在しないこと、および／または現在演算装置の第2の属性情報と一致する履歴演算装置が存在しないことに応答して、第1の属性情報、第2の属性情報および入力データに必要な演算子に基づいて、候補演算子生成方式の損失情報を取得し、損失情報に基づいて候補演算子生成方式の性能評価結果を生成する。

【0086】

選択可能に、性能評価モデルを予め構築し、性能評価モデルを訓練し、第1の属性情報、第2の属性情報および入力データに必要な演算子を性能評価モデルに入力して、候補演算子生成方式の性能評価結果を出力することができる。

【0087】

さらに、候補演算子生成方式の性能評価結果を生成した後、性能評価結果に基づいて候補演算子生成方式からターゲット演算子生成方式を出力する必要がある。

【0088】

出力された性能評価結果に対応する履歴性能評価結果が存在しないため、本開示の実施例では、性能評価結果を取得した後、決定されたターゲット演算子生成方式の性能評価結果を現在の入力データの第1の属性情報および現在の演算装置の第2の属性情報にバインディングして、新しい履歴性能評価結果として記憶することにより、データライブラリを更新し、データライブラリの完全性を向上させ、演算子生成方式のスケジューリング効率を向上させることができる。

【0089】

以上により、本開示の実施例のデータ処理方法は、様々な方式に基づいて、性能評価結果を取得することができ、選択可能に、履歴性能評価結果には、履歴入力データの第1の属性情報が第1の属性情報と一致し、且つ履歴演算装置の第2の属性情報が第2の属性情報と一致するものが含まれる場合、性能評価結果を直接取得することができる。選択可能に、履歴性能評価結果にはターゲット性能評価結果が含まれていない場合、候補演算子生成方式の損失情報を取得し、損失情報に基づいて候補演算子生成方式の性能評価結果を生成し、性能評価結果を新しい履歴性能評価結果として記憶する。これにより、本開示は複数の方式に基づいて性能評価結果を取得することができるため、性能評価結果を取得する精度および信頼性が向上させることができ、また、性能評価結果に基づいて対応する演算子生成方式を選択することができるため、演算子開発時の作業量を低減することができる。

【0090】

1つの可能な実現形態として、図5に示すように、上記実施例のもとに、性能評価結果に基づいて、候補演算子生成方式からターゲット演算子生成方式を選択するステップは、具体的には、以下のステップS501～S502を含むことができる。

【0091】

S501において、ターゲット演算子生成方式を選択するためのターゲット性能指標を決定する。

【0092】

なお、ターゲット性能指標は、消費電力、グラフィックメモリ使用量、かかる時間のうちの少なくとも1つのであってもよい。データの演算ニーズに応じて、複数の性能指標からターゲット性能指標を決定することができる。

【0093】

S502において、ターゲット性能指標と性能評価結果とに基づいて、候補演算子生成方式からターゲット演算子生成方式を選択する。

【0094】

本開示の実施例では、性能評価結果には1つまたは複数の性能指標の評価結果が含まれ、ターゲット性能指標を取得した後、性能評価結果のうちのターゲット性能指標の指標評価結果を決定することができ、ターゲット性能指標の指標評価結果に基づいて、候補演算子生成方式からターゲット演算子生成方式を選択し、例えば、ターゲット性能指標がかかる時間である場合、指標評価結果におけるかかる時間が最も小さい性能評価結果に対応する候補演算子生成方式をターゲット演算子生成方式として決定することができる。また、ターゲット性能指標がグラフィックメモリ使用量である場合、グラフィックメモリ使用量が最も少ない性能評価結果に対応する候補演算子生成方式をターゲット演算子生成方式として決定することができる。例えば、A候補演算子生成方式とB候補演算子生成方式があり、B候補演算子生成方式の性能評価結果にはターゲット性能指標が含まれておらず、A候補演算子生成方式の性能評価結果にはターゲット性能指標が含まれる場合、A候補演算子生成方式を直接ターゲット演算子生成方式として選択することができる。

【0095】

1つの可能な実現形態として、図6に示すように、上記実施例のもとに、ターゲット性能指標と性能評価結果とに基づいて、候補演算子生成方式からターゲット演算子生成方式

10

20

30

40

50

を選択するステップは、具体的には、以下のステップ S 6 0 1 ～ S 6 0 3 を含むことができる。

【0096】

S 6 0 1 において、性能選好選択ポリシーを選択し、性能選好選択ポリシーに基づいて、ターゲット性能指標を決定する。

【0097】

ここで、性能選好選択ポリシーとは、演算ニーズに応じて予め決定された、ターゲット演算子生成方式を選択するポリシーを指す。

【0098】

さらに、性能選好選択ポリシーを取得した後、性能選好選択ポリシーから選択性能評価結果の1つまたは複数のターゲット性能指標を決定することができる。

10

【0099】

S 6 0 2 において、性能評価結果から、ターゲット性能指標にマッチングする指標評価結果を選択する。

【0100】

本開示の実施例では、性能評価結果には1つまたは複数の性能指標の結果が含まれ、例えば、消費電力、グラフィックメモリ使用量、かかる時間のうちの1つまたは複数が含まれる。ターゲット性能指標を取得した後、性能評価結果から、ターゲット性能指標にマッチングする指標評価結果を決定することができる。

【0101】

例えば、取得したターゲット指標が消費電力である場合、性能評価結果から消費電力指標の評価結果を選択することができる。

20

【0102】

S 6 0 3 において、指標評価結果に基づいて、候補演算子生成方式からターゲット演算子生成方式を選択する。

【0103】

本開示の実施例では、指標評価結果を取得した後、指標評価結果に基づいて、候補演算子生成方式からターゲット演算子生成方式を選択することができる。

【0104】

例えば、性能選好ポリシーが消費電力である場合、演算子組み合わせ実現方式の消費電力が演算子コンパイラ実現方式より低い時、演算子組み合わせ実現方式のかかる時間が演算子コンパイラ実現方式より長くても、演算子組み合わせ実現方式をターゲット実現方式とする。

30

【0105】

以上により、本開示の実施例データ処理方法は、性能選好選択ポリシーを選択し、性能選好選択ポリシーに基づいて、ターゲット性能指標を決定し、性能評価結果からターゲット性能指標にマッチングする指標評価結果を選択し、その後、指標評価結果に基づいて、候補演算子生成方式からターゲット演算子生成方式を選択することができる。これにより、予め演算ニーズに応じて性能選好選択ポリシーを取得し、指標評価結果に基づいてターゲット演算子生成方式を選択することができる。これにより、演算子生成方式を包括的にスケジューリングすることができ、演算子実現プロセスは、より低い消費電力またはより短いかかる時間またはより少ないリソース占有量などの利点を有する。

40

【0106】

1つの可能な実現形態として、図7に示すように、上記実施例のもとに、入力データに必要な演算子に含まれるサブ演算子に基づいて、演算子を生成するステップは、具体的には、以下のステップ S 7 0 1 ～ S 7 0 2 を含むことができる。

【0107】

以下、ターゲット演算子生成方式が演算子組み合わせ実現方式および演算子コンパイラ実現方式である場合について説明する。

【0108】

ターゲット演算子生成方式が演算子組み合わせ実現方式である場合について説明する。

50

【0109】

S701において、ターゲット演算子生成方式が演算子組み合わせ実現方式であることに応答して、入力データの第1の属性情報に基づいて、必要な演算子に含まれる少なくとも1つの第1のサブ演算子と、第1のサブ演算子間の組み合わせ順序とを取得する。

【0110】

なお、演算子組み合わせ実現方式とは、一定の組み合わせ順序に従って、サブ演算子を組み合わせさせて複雑な演算子を取得することである。

【0111】

なる入力データの属性情報は異なる場合が多く、異なる属性情報は入力データの実行ニーズを反映することができ、そして入力データに対応する演算子を決定する。本開示の実施例では、入力データの第1の属性情報に基づいて、入力データの演算ニーズを決定することができ、演算ニーズに応じて入力データの演算子を決定することができ、演算子を取得した後に、演算子組み合わせ実現方式に基づいて、当該演算子に含まれる第1のサブ演算子、および第1のサブ演算子間の組み合わせ順序を決定することができる。

10

【0112】

例えば、演算子Aについて、取得された第1のサブ演算子はサブ演算子1、サブ演算子2、サブ演算子3、サブ演算子4、およびサブ演算子5であり、取得された第1のサブ演算子間の組み合わせ順序は、サブ演算子4、サブ演算子5、サブ演算子2、サブ演算子3、サブ演算子1である。

20

【0113】

S702において、組み合わせ順序に基づいて、第1のサブ演算子を順次組み合わせて、入力データに対応する演算子を取得する。

【0114】

本開示の実施例では、ターゲット演算子生成方式が演算子組み合わせ実現方式である場合、第1のサブ演算子の組み合わせ順序に従って、第1のサブ演算子を順次組み合わせて、入力データに対応する演算子を取得することができる。

【0115】

以上により、本開示の実施例に係るデータ処理方法は、ターゲット演算子生成方式が演算子組み合わせ実現方式であることに応答して、入力データの第1の属性情報に基づいて、演算子に含まれる少なくとも1つの第1のサブ演算子、および第1のサブ演算子間の組み合わせ順序を取得し、組み合わせ順序に基づいて、第1のサブ演算子を順次組み合わせて、入力データに対応する演算子を取得する。これにより、本開示は、サブ演算子を順次組み合わせて、入力データに対応する演算子を取得することができ、ハードウェアインターフェース作業の量が低減される。

30

【0116】

入力データに対応する演算子を取得した後、演算装置により演算子に基づいて入力データを演算する時、組み合わせ順序に応じて、演算子に含まれる第1のサブ演算子を順次実行する必要があり、即ち実行順序は、サブ演算子4ーサブ演算子5ーサブ演算子2ーサブ演算子3ーサブ演算子1であり、最後には、演算結果を出力する。

【0117】

ターゲット演算子生成方式が演算子コンパイラ実現方式である場合について説明する。

40

【0118】

S801において、ターゲット演算子生成方式が演算子コンパイラ実現方式であることに応答して、入力データの第1の属性情報に基づいて、必要な演算子と、必要な演算子に含まれる少なくとも1つの第2のサブ演算子と、第2のサブ演算子のコンパイル順序とを取得する。

【0119】

ここで、演算子コンパイラ実現方式とは、一定のコンパイル順序に従って、サブ演算子をコンパイルすることで複雑な演算子を取得することである。

【0120】

50

異なる入力データの属性情報は異なる場合が多く、異なる属性情報は入力データの実行ニーズを反映することができ、そして入力データに対応する演算子を決定する。本開示の実施例では、入力データの第1の属性情報に基づいて、入力データの演算ニーズを決定することができ、演算ニーズに基づいて入力データの演算子を決定することができ、演算子を取得した後に、演算子コンパイラ実現方式に基づいて、当該演算子に含まれる第2のサブ演算子、および第2のサブ演算子間の組み合わせ順序を決定することができる。

【0121】

例えば、演算子Bについて、取得された第2のサブ演算子はサブ演算子6、サブ演算子7、サブ演算子8、サブ演算子9、およびサブ演算子10であり、取得された第2のサブ演算子間のコンパイル順序は、サブ演算子9、サブ演算子10、サブ演算子7、サブ演算子6、サブ演算子8である。

【0122】

S802において、すべての第2のサブ演算子に対してチューニング処理を行って、チューニングされたターゲット第2のサブ演算子を取得する。

【0123】

1つの可能な実現形態として、ターゲットチューニングポリシーを取得し、ここで、ターゲットチューニングポリシーは、第2のサブ演算子についての融合ポリシーを含むことができ、さらに、ターゲットチューニングポリシーに基づいて、すべての第2のサブ演算子をチューニング処理する。

【0124】

選択可能に、第1の属性情報に基づいて、入力データのデータ形状を決定し、データ形状に基づいて、対応するターゲットチューニングポリシーを取得することができる。

【0125】

本開示では、第1の属性情報に基づいて、入力データのデータ形状を決定する具体的な方式が限定されず、実際の状況に応じて選択することができる。

【0126】

選択可能に、入力データinput1に対して、入力データinput1と第1の属性情報との間のマッピング関係を照会することで、入力データの形状を決定することができる。

【0127】

本開示の実施例では、データ形状を取得した後、対応するターゲットチューニングポリシーを取得することができる。例えば、演算子をコンパイルして、装置全体の実行可能なコードに最適化することができる。

【0128】

さらに、ターゲットチューニングポリシーを取得した後、すべての第2のサブ演算子をチューニング処理することができる。

【0129】

なお、本開示では、すべての第2のサブ演算子をチューニング処理する具体的なプロセスが限定されず、実際の状況に応じて選択することができる。

【0130】

選択可能に、自動または手動ですべての第2のサブ演算子をチューニング処理することができ、即ち、予め設定されたチューニングルールまたはチューニングテンプレートに基づいてすべての第2の演算子をチューニング処理し、チューニングされたターゲット第2のサブ演算子を取得することができる。

【0131】

選択可能に、ターゲットチューニングポリシーは第2のサブ演算子についての融合ポリシーを含むことができ、すべての第2のサブ演算子に対して演算子融合を行うことができ、さらには、対応するターゲットチューニングポリシーを取得した後、ターゲットチューニングポリシーに基づいてすべての第2のサブ演算子に対して演算子融合を行って、融合サブ演算子を取得することができる。

10

20

30

40

50

【0132】

さらに、融合された融合サブ演算子に対して演算子グルーピングおよび演算子スケジューリングを行って、ターゲット第2のサブ演算子を取得する。

【0133】

なお、融合された第2の演算子を取得した後、第2の演算子に対応する入力データの形状と演算装置とに基づいてグルーピングおよび演算子スケジューリングを行って、ターゲット第2のサブ演算子を取得することができる。

【0134】

S803において、ターゲット第2のサブ演算子間のコンパイル順序に基づいて、ターゲット第2のサブ演算子を順次コンパイルして、入力データに対応する演算子を生成する。

【0135】

以上により、本開示の実施例に係るデータ処理方法は、性能選好選択ポリシーに基づいて、演算子組み合わせ実現方式または演算子コンパイラ実現方式を選択し、演算子組み合わせ実現方式と演算子コンパイラ実現方式とに基づいて、演算装置により演算子に従って入力データを演算することができる。したがって、ターゲット第2のサブ演算子間のコンパイル順序に基づいて、ターゲット第2のサブ演算子を順次コンパイルして、入力データに対応する演算子を生成することができ、演算子が装置の命令実行を起動させる時のオーバーヘッドを回避することができる。

【0136】

さらに、演算子を取得した後、演算装置により演算子に従って入力データの演算を行って、演算結果を取得することができる。

【0137】

1つの可能な実現形態として、図9に示すように、上記実施例のもとに、演算装置により演算子に従って入力データの演算を行って、演算結果を取得するステップは、具体的に、以下のステップS901～S902を含むことができる。

【0138】

S901において、入力データの演算子に基づいて、入力データに対応するディープニューラルネットワークの演算子セットを生成する。

【0139】

ここで、ディープニューラルネットワーク（Deep Neural Networks、DNNと略称する）は機械学習（Machine Learning、MLと略称する）分野における技術である。

【0140】

なお、データ処理タスクを受信した後、データ処理タスクから入力データを抽出することができ、データ処理タスクの多様性により、入力データも多様性を有し、同様に、入力データに対応する演算子も多様性を有する。したがって、入力データの演算子に基づいて、入力データに対応するディープニューラルネットワークの演算子セットを生成することができる。

【0141】

S902において、演算子セットを演算装置に配布してディープニューラルネットワークを構築し、ディープニューラルネットワークにより入力データの演算を行って演算結果を取得する。

【0142】

本開示の実施例では、入力データに対応するディープニューラルネットワークの演算子セットを生成した後、演算子セットを演算装置に配布してディープニューラルネットワークを構築することができる。

【0143】

例えば、入力データは画像データであってもよく、ローカル畳み込み－正規化－非線形化ネットワーク（Conv-IN-Relu）を構築することができ、入力データはオーディオデータであってもよく、ウェーブレットニューロンネットワーク（Wavelet

10

20

30

40

50

Neural Network、WNNと略称する)を構築することができ、入力データはテキストデータであってもよく、畳み込みリカレントニューロンネットワーク(Convolutional Recurrent Neural Network、CRNNと略称する)を構築することができる。

【0144】

さらに、ディープニューラルネットワークを構築した後、ディープニューラルネットワークにより入力データの演算を行って演算結果を取得することができる。異なるディープニューラルネットワークは入力データの演算を行って、異なる演算結果を取得することができる。例えば、入力データが画像データである時、構築されたディープニューラルネットワークはターゲット検出、画像分割のためのニューロンネットワークであってもよい。これに対して、演算結果は出力されたターゲット検出フレーム、画像分割結果、検出されたターゲットタイプなどであってもよい。また、入力データがオーディオデータである場合、構築されたディープニューラルネットワークは、話者タイプ認識、オーディオ認識などに用いられるニューロンネットワークであってもよく、これに対して、演算結果は、出力された話者タイプ、オーディオに対応するテキストなどであってもよい。また、入力データはテキストデータであってもよく、構築されたディープニューラルネットワークはテキスト抽出ネットワーク、テキスト生成ネットワークなどであってもよく、これに対して、演算結果は、抽出されたエンティティ情報を出力すること、入力データに基づいて生成された文章などであってもよい。

【0145】

同時に、演算プロセスにおける効率および精度を向上させるために、演算プロセスを監視し、演算プロセスに異常が発生すると、演算を停止し且つ異常ログを生成することができる。ここで、異常ログは、演算子を改めてスケジューリングするために使用することができる。

【0146】

選択可能に、演算プロセスに異常が発生する時、演算装置の第2の属性情報に基づいて、新たな第1の演算装置をスケジューリングし、新たな第1の演算装置により演算子に従って入力データの演算を行って演算結果を取得することができる。

【0147】

選択可能に、演算プロセスに異常が発生する時、入力データの第1の属性情報に基づいて、新たな第2の演算装置をスケジューリングし、第2の演算装置の第2の属性情報を取得し、さらに、第1の属性情報と第2の属性情報とに基づいて候補演算子生成方式からターゲット演算子生成方式を選択するステップおよびその後のステップに戻ることができる。

【0148】

以上により、本開示の実施例に係るデータ処理方法は、入力データの演算子に基づいて、入力データに対応するディープニューラルネットワークの演算子セットを生成し、演算子セットを演算装置に配布してディープニューラルネットワークを構築し、ディープニューラルネットワークにより入力データの演算を行って演算結果を取得できるとともに、演算プロセスの異常を監視することができ、演算プロセスに異常が発生する時に、新たな第1の演算装置により演算子に従って入力データの演算を行って演算結果を取得ことができ、また、入力データの第1の属性情報に基づいて、新たな第2の演算装置をスケジューリングすることができる。これにより、演算装置においてディープニューラルネットワークを構築して、入力データの演算を行って演算結果を取得ことができ、入力データ演算プロセスにおける効率および精度を向上させることができる。

【0149】

以下、本開示によって提供されるデータ処理のフレームを説明する。

【0150】

図10に示すように、本開示のデータ処理フレームは、データ処理装置を備え、当該データ処理装置は上記実施例にかかるデータ処理方法を実行することができ、即ち、入力データの第1の属性情報と演算装置の第2の属性情報とに基づいて、演算子生成方式ライブ

ラリから1つのターゲット演算子生成方式をスケジューリングすることができ、ここで、演算子生成方式ライブラリは少なくとも演算子組み合わせ実現方式と演算子コンパイラ実現方式とを含み、入力データに対応するいずれか1つの演算子に対してスケジューリングされたターゲット演算子生成方式は、演算子組み合わせ実現方式および演算子コンパイラ実現方式の一方である。入力データに対応する演算子は複数のサブ演算子を含み、演算子ライブラリから演算子に対応するサブ演算子を選択し、その後、ターゲット演算子生成方式でサブ演算子を処理して、入力データに対応する演算子を生成する必要がある。さらに、演算子生成方式を取得した後、演算子生成方式を演算装置にロードし、演算装置により演算子に従って入力データを演算することができる。

【0151】

以下、本開示に係るデータ処理方法の流れを説明する。

【0152】

図11に示すように、入力データがinput1である場合、演算装置の加速ライブラリバージョンをCUDNNに選定し、ここで、input1に対応する演算子はフル接続層(Fully Connection、FC)演算子と正規化(softmax)演算子とを含み、ローカルまたはリモートで性能テスト結果を取得することができ、性能テスト結果が選択可能である場合、ターゲット演算子生成方式を直接スケジューリングすることができ、その後、FC演算子およびsoftmax演算子を生成し、演算装置によりこの2つの演算子に従ってinput1の演算結果output1を出力する。

【0153】

図12に示すように、入力データがinput2であり、且つinput2のデータ形状がinput1のデータ形状と異なる場合、改めて演算子生成方式のスケジューリングを行う必要があり、演算装置の加速ライブラリバージョンをCUDAに選定し、input2に対応する演算子はFC演算子とsoftmax演算子とを含み、ローカルまたはリモートで性能テスト結果を取得することができ、性能テスト結果が選択可能である場合、演算装置により演算結果output2を直接出力する。

【0154】

図13に示すように、入力データがinput3であり、且つinput3のデータ形状がinput1のデータ形状と同じである場合、演算装置の加速ライブラリバージョンをCUDAに選定し、input3に対応する演算子はFC演算子およびsoftmax演算子であり、ローカルまたはリモートで性能テスト結果を取得することができ、性能テスト結果が選択可能である場合、input1と同じ演算子生成方式を直接取得し、且つ演算装置により演算結果output3を出力することができる。

【0155】

図14は、本開示に係るデータ処理方法を実現するためのデータ処理装置1400のブロック図である。図14に示すように、データ処理装置1400は、取得モジュール1501、選択モジュール1402、生成モジュール1403および演算モジュール1404を備える。

【0156】

取得モジュール1401は、演算子に対応する入力データの第1の属性情報を取得し、且つ前記入力データに対応する演算装置の第2の属性情報を取得するように構成される。

選択モジュール1402は、前記第1の属性情報と前記第2の属性情報とに基づいて、候補演算子生成方式からターゲット演算子生成方式を選択するように構成される。

生成モジュール1403は、前記ターゲット演算子生成方式に基づいて、演算子ライブラリから入力データに必要な演算子に含まれるサブ演算子を決定して、前記演算子を生成するように構成される。

演算モジュール1404は、前記演算装置により前記演算子に従って前記入力データの演算を行って、演算結果を取得するように構成される。

【0157】

選択モジュール1402は、さらに、前記第1の属性情報と前記第2の属性情報とに基

10

20

30

40

50

づいて、前記候補演算子生成方式の性能評価を行って、対応する性能評価結果を取得し、前記性能評価結果に基づいて、前記候補演算子生成方式から前記ターゲット演算子生成方式を選択するように構成される。

【0158】

選択モジュール1402は、さらに、前記演算子の前記候補演算子生成方式に対応する履歴性能評価結果を取得し、前記第1の属性情報と前記第2の属性情報に基づいて、前記履歴性能評価結果からターゲット性能評価結果を前記性能評価結果として選択するように構成される。

【0159】

選択モジュール1402は、さらに、前記履歴性能評価結果に基づいて、対応する履歴入力データと履歴演算装置とを取得し、前記履歴性能評価結果における前記履歴入力データの第1の属性情報および前記入力データの第1の属性情報と、前記履歴演算装置の第2の属性情報および前記演算装置の第2の属性情報とに基づいて、前記履歴性能評価結果からターゲット性能評価結果を前記性能評価結果として選択するように構成される。

【0160】

選択モジュール1402は、さらに、前記履歴入力データの第1の属性情報と前記入力データの第1の属性情報とを比較し、前記履歴演算装置の第2の属性情報と前記演算装置の第2の属性情報とを比較し、前記履歴性能評価結果から、前記入力データの第1の属性情報と一致し且つ前記演算装置の第2の属性情報と一致するターゲット性能評価結果を前記性能評価結果として選択するように構成される。

【0161】

選択モジュール1402は、さらに、前記性能評価結果に対応する候補演算子生成方式を前記ターゲット演算子生成方式として決定するように構成される。

【0162】

選択モジュール1402は、さらに、前記ターゲット演算子生成方式に対応する性能評価結果を、前記第1の属性情報および前記第2の属性情報にバインディングして、1つの新しい履歴性能評価結果として記憶するように構成される。

【0163】

選択モジュール1402は、さらに、性能選好選択ポリシーを取得し、前記性能選好選択ポリシーに基づいて、ターゲット指標を決定し、前記性能評価結果から、前記ターゲット性能指標にマッチングする指標評価結果を選択し、前記指標評価結果に基づいて、前記候補演算子生成方式から前記ターゲット演算子生成方式を選択するように構成される。

【0164】

生成モジュール1403は、さらに、前記ターゲット演算子生成方式が演算子組み合わせ実現方式であることに応答して、前記入力データの第1の属性情報に基づいて、必要な演算子と、前記必要な演算子に含まれる少なくとも1つの第1のサブ演算子と、前記第1のサブ演算子間の組み合わせ順序とを取得し、前記組み合わせ順序に基づいて、前記第1のサブ演算子を前記演算子に順次組み合わせるように構成される。

【0165】

生成モジュール1403は、さらに、前記ターゲット演算子生成方式が演算子コンパイラ実現方式であることに応答して、前記入力データの第1の属性情報に基づいて、必要な演算子と、前記必要な演算子に含まれる少なくとも1つの第2のサブ演算子と、前記第2のサブ演算子のコンパイル順序とを取得し、すべての前記第2のサブ演算子に対してチューニング処理を行って、チューニングされたターゲット第2のサブ演算子を取得し、前記コンパイル順序に基づいて、前記ターゲット第2のサブ演算子を順次コンパイルして、前記演算子を生成するように構成される。

【0166】

生成モジュール1403は、さらに、ターゲットチューニングポリシーを取得し、前記ターゲットチューニングポリシーに基づいてすべての前記第2のサブ演算子に対して演算子融合を行って、融合サブ演算子を取得して、前記融合サブ演算子に対して演算子グルー

10

20

30

40

50

ピングおよび演算子スケジューリングを行って、前記ターゲット第2のサブ演算子を取得するように構成される。

【0167】

生成モジュール1403は、さらに、前記第1の属性情報に基づいて、前記入力データのデータ形状を決定し、前記データ形状に基づいて、対応する前記ターゲットチューニングポリシーを取得するように構成される。

【0168】

演算モジュール1404は、さらに、前記入力データの前記演算子に基づいて、前記入力データに対応するディープニューラルネットワークの演算子セットを生成し、前記演算子セットを前記演算装置に配布して前記ディープニューラルネットワークを構築し、前記ディープニューラルネットワークにより前記入力データを演算して前記演算結果を取得するように構成される。

【0169】

演算モジュール1404は、さらに、前記演算プロセスを監視し、前記演算プロセスの異常にตอบสนองして、演算を停止し且つ異常ログを生成するように構成される。

【0170】

演算モジュール1404は、さらに、前記演算装置の第2の属性情報に基づいて、新たな第1の演算装置をスケジューリングし、前記新たな第1の演算装置により前記演算子に従って前記入力データの演算を行って、演算結果を取得するように構成される。

【0171】

演算モジュール1404は、さらに、前記入力データの第1の属性情報に基づいて、新たな第2の演算装置をスケジューリングし、前記第2の演算装置の第2の属性情報を取得し、さらに、前記第1の属性情報と前記第2の属性情報とに基づいて候補演算子生成方式からターゲット演算子生成方式を選択するステップおよびその後のステップに戻るように構成される。

【0172】

取得モジュール1401は、さらに、データ処理タスクを受信し、前記データ処理タスクから前記入力データを抽出するように構成され、前記入力データが、画像データ、オーディオデータ、およびテキストデータのうちの1つである。

【0173】

以上により、本開示の実施例に係るデータ処理装置は、入力データの第1の属性情報を取得し、且つ入力データに対応する演算装置の第2の属性情報を取得し、第1の属性情報と第2の属性情報とに基づいて、候補演算子生成方式からターゲット演算子生成方式を選択し、その後、ターゲット演算子生成方式に基づいて、演算子ライブラリから入力データに必要な演算子に含まれるサブ演算子を決定して、演算子を生成し、最後に、演算装置により演算子に従って入力データの演算を行って、演算結果を取得することができる。本開示の実施例では、データの第1の属性情報と演算装置の第2の属性情報とに基づいて、演算子生成方式の自動的なスケジューリングを行うことができ、各演算子生成方式の優勢を十分に生かし、ハードウェアの性能を十分に生かすことができる。

【0174】

本開示の技術案では、関わるユーザ個人情報の収集、記憶、使用、加工、伝送、提供および公開などの処理のすべては、関連する法律および法規の規定に合致し、且つ公序良俗に違反しない。

【0175】

本開示の実施例によれば、本開示は、電子機器、読み取り可能な記憶媒体およびコンピュータプログラムを提供する。

【0176】

図15は、本開示の実施例を実現するための例示的な電子機器1300の概略ブロック図を示す。電子機器1300は、ラップトップコンピュータ、デスクトップコンピュータ、ワークステーション、パーソナルデジタルアシスタント、サーバ、ブレードサーバ、メ

10

20

30

40

50

インフレーム、及び他の適切なコンピュータなどの様々な形式のデジタルコンピュータを表すことを目的とする。電子機器は、携帯情報端末、携帯電話、スマートフォン、ウェアラブルデバイス、他の同様のコンピューティングデバイスなどの様々な形式のモバイルデバイスを表すこともできる。本明細書で示されるコンポーネント、それらの接続と関係、及びそれらの機能は単なる例であり、本明細書の説明及び／又は要求される本開示の実現を制限することを意図したものではない。

【0177】

図15に示すように、電子機器1500は、リードオンリーメモリ（ROM）1502に記憶されるコンピュータプログラムまたは記憶ユニット1508からランダムアクセスメモリ（RAM）1503にロードされるコンピュータプログラムに基づいて様々な適切な動作と処理を行うことができる計算ユニット1501を備える。RAM 1503では、電子機器1500の操作に必要な様々なプログラムおよびデータを記憶することができる。計算ユニット1501、ROM 1502およびRAM 1503はバス1504を介して互いに接続される。入力／出力（I／O）インターフェース1505もバス1504に接続される。

【0178】

電子機器1500内の複数の部品はI／Oインターフェース1505に接続され、キーボード、マウスなどの入力ユニット1506と、様々なタイプのディスプレイ、スピーカーなどの出力ユニット1507と、磁気ディスク、光ディスクなどの記憶ユニット1508と、ネットワークカード、モデム、無線通信送受信機などのおよび通信ユニット1509と、を備える。通信ユニット1509は、電子機器1500がインターネットのようなコンピュータネットワークおよび／または様々な電信ネットワークを介して他のデバイスと情報／データを交換すること許容する。

【0179】

計算ユニット1501は処理と計算能力を備える様々な汎用および／または専用処理コンポーネントであってもよい。計算ユニット1501のいくつかの例は、中央処理装置（CPU）、グラフィック処理装置（GPU）、様々な専有の人工知能（AI）計算チップ、機械学習モデルアルゴリズムを実行する様々な計算ユニット、デジタル信号プロセッサ（DSP）、および如何なる適切なプロセッサ、コントローラ、マイクロコントローラなどを含むが、これらに限定されない。計算ユニット1501は上記で説明された各方法と処理を実行し、例えば図1～図13に示すデータ処理方法を実行する。例えば、いくつかの実施例では、車線変更制御方法はコンピュータソフトウェアプログラムとして実現することができ、それは有形的に機器読み取り可能な媒体、例えば記憶ユニット1508に含まれる。いくつかの実施例では、コンピュータプログラムの一部または全部はROM 1502および／または通信ユニット1509を介して電子機器1500にロードおよび／またはインストールすることができる。コンピュータプログラムがRAM 1503にロードされ且つ計算ユニット1501によって実行される時、上記で説明されたデータ処理方法の1つまたは複数のステップを実行することができる。予備的に、他の実施例では、計算ユニット1501は他の任意の適切な方式（例えば、ファームウェアにより）により、データ処理方法を実行するように構成されてもよい。

【0180】

本文の以上で説明されたシステムと技術の様々な実施形態は、デジタル電子回路システム、集積回路システム、フィールド・プログラマブル・ゲート・アレイ（FPGA）、特定用途向け集積回路（ASIC）、専用標準製品（ASSP）、システムオンチップ（SOC）、複雑なプログラマブル論理デバイス（CPLD）、コンピュータハードウェア、ファームウェア、ソフトウェア、および／またはそれらの組み合わせで実現することができる。これらの様々な実施形態は、1つまたは複数のコンピュータプログラム内で実行することを含むことができ、当該1つまたは複数のコンピュータプログラムは、少なくとも1つのプログラマブルプロセッサを備えるプログラマブルシステムにおいて実行および／または解釈することができ、当該プログラマブルプロセッサは専用または汎用プログラマ

ブルプロセッサであってもよく、記憶システム、少なくとも1つの入力装置、および少なくとも1つの出力装置からデータと命令を受信することができ、且つデータと命令を当該記憶システム、当該少なくとも1つの入力装置、および当該少なくとも1つの出力装置に送信する。

【0181】

本開示の方法を実施するプログラムコードは1つまたは複数のプログラミング言語の任意の組み合わせで書くことができる。これらのプログラムコードは汎用コンピュータ、専用コンピュータまたは他のプログラマブル車線変更制御装置のプロセッサまたはコントローラに提供することができ、これにより、プログラムコードがプロセッサまたはコントローラによって実行される際に、フローチャートおよび／またはブロック図で規定される機能／操作は実施される。プログラムコードは完全に機器で実行するか、一部が機器で実行するか、独立したソフトウェアパッケージとして部分的に機器で実行され且つ部分的にリモート機器で実行されるか、または完全にリモート機器またはサーバで実行される。

【0182】

本開示の文脈では、機器読み取り可能な媒体は有形の媒体であってもよく、命令実行システム、装置またはデバイスで使用するプログラム、または命令実行システム、装置またはデバイスに合わせて使用するプログラムを含むまたは記憶することができる。機器読み取り可能な媒体は機器読み取り可能な信号媒体または機器読み取り可能な記憶媒体である。機器読み取り可能な媒体は電子的、磁性的、光学的、電磁的、赤外線のもの、または半導体システム、装置またはデバイス、または上記内容の如何なる適切な組み合わせを含むことができるが、これらに限定されない。機器読み取り可能な記憶媒体のより具体的な例は、1つまたは複数のラインに基づく電氣的接続、ポータブルコンピュータディスク、ハードディスク、ランダムアクセスメモリ（RAM）、リードオンリーメモリ（ROM）、消去可能なプログラマブルリードオンリーメモリ（EPROMまたはフラッシュメモリ）、フラッシュ、コンパクトディスクリードオンリーメモリ（CD-ROM）、光学記憶デバイス、磁気記憶デバイス、または上記内容の如何なる適切な組み合わせを含むことができる。

【0183】

ユーザとのインタラクションを提供するために、ここで説明されているシステム及び技術をコンピュータ上で実施することができ、当該コンピュータは、ユーザに情報を表示するためのディスプレイ装置（例えば、CRT（陰極線管）又はLCD（液晶ディスプレイ）モニタ）と、キーボード及びポインティングデバイス（例えば、マウス又はトラックボール）とを有し、ユーザは、当該キーボード及び当該ポインティングデバイスによって入力をコンピュータに提供することができる。他の種類の装置も、ユーザとのインタラクションを提供することができ、例えば、ユーザに提供されるフィードバックは、任意の形式のセンシングフィードバック（例えば、視覚フィードバック、聴覚フィードバック、又は触覚フィードバック）であってもよく、任意の形式（音響入力と、音声入力、または、触覚入力とを含む）でユーザからの入力を受信することができる。

【0184】

ここで説明されるシステム及び技術は、バックエンドコンポーネントを備えるコンピューティングシステム（例えば、データサーバとする）、又はミドルウェアコンポーネントを備えるコンピューティングシステム（例えば、アプリケーションサーバ）、又はフロントエンドコンポーネントを備えるコンピューティングシステム（例えば、グラフィカルユーザインタフェース又はウェブブラウザを有するユーザコンピュータ、ユーザは、当該グラフィカルユーザインタフェース又は当該ウェブブラウザによってここで説明されるシステム及び技術の実施形態とインタラクションできる）、又はこのようなバックエンドコンポーネントと、ミドルウェアコンポーネントと、フロントエンドコンポーネントのいずれかの組み合わせを備えるコンピューティングシステムで実行することができる。いずれかの形態又は媒体のデジタルデータ通信（例えば、通信ネットワーク）によってシステムのコンポーネントを相互に接続することができる。通信ネットワークの例は、ローカルエリ

アネットワーク（LAN）、ワイドエリアネットワーク（WAN）、インターネットおよびブロックチェーンネットワークを含む。

【0185】

コンピュータシステムはクライアントとサーバを備えることができる。クライアントとサーバは、一般に、互いに離れており且つ通信ネットワークを介して対話する。対応するコンピュータで実行され且つ互いにクライアントーサーバ関係を有するコンピュータプログラムを介してクライアントとサーバとの関係を生成する。サーバはクラウドサーバであってもよいし、分散システムのサーバであってもよいし、またはブロックチェーンネットワークと組み合わせられるサーバであってもよい。

【0186】

本開示の実施例により、本開示はコンピュータプログラムをさらに提供し、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、本開示の上記実施例に係るデータ処理方法を実現する。

【0187】

なお、以上に示す様々な形式のフローを使用して、ステップを改めて並べ替え、追加または削除することができることを理解されたい。例えば、本開示に記載の各ステップは、並列に実行されてもよいし、順次的に実行されてもよいし、異なる順序で実行されてもよいが、本開示で開示されている技術案の所望の結果を実現することができれば、本明細書では限定されない。

【0188】

上記具体的な実施形態は、本開示に対する保護範囲を限定するものではない。当業者であれば、設計要件と他の要因に応じて、様々な修正、組み合わせ、一部の組み合わせ及び代替を行うことができる。本開示の精神と原則内で行われる任意の修正、同等の置換、及び改善などは、いずれも本開示の保護範囲内に含まれるべきである。

10

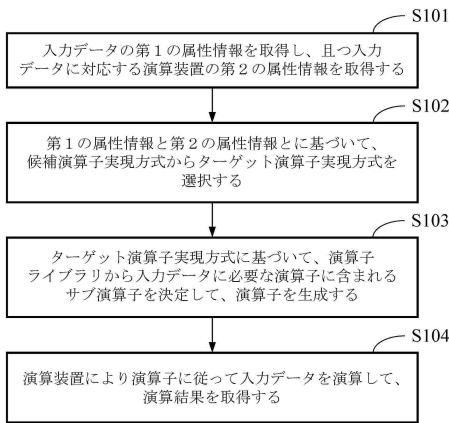
20

30

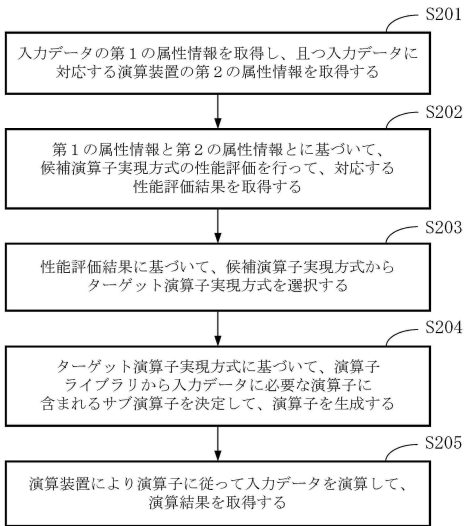
40

50

【図面】
【図 1】



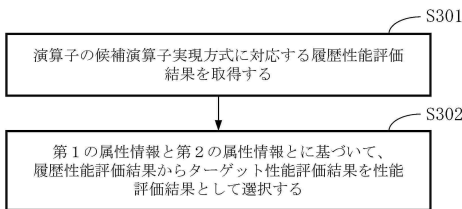
【図 2】



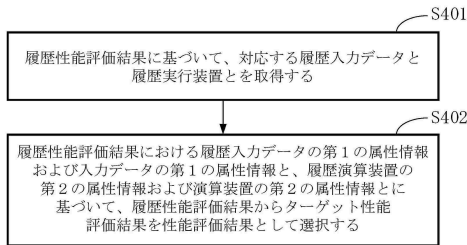
10

20

【図 3】



【図 4】

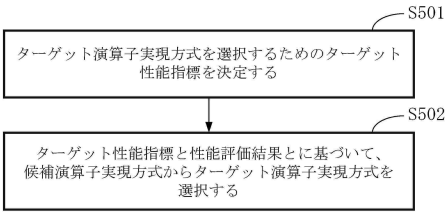


30

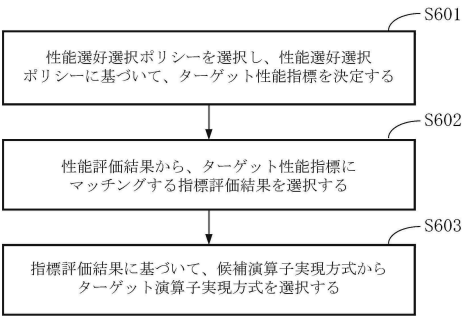
40

50

【図 5】

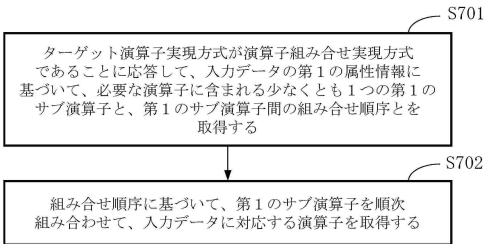


【図 6】

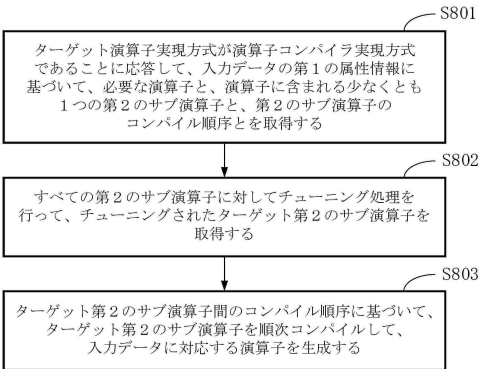


10

【図 7】



【図 8】



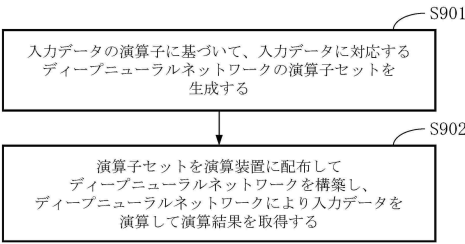
20

30

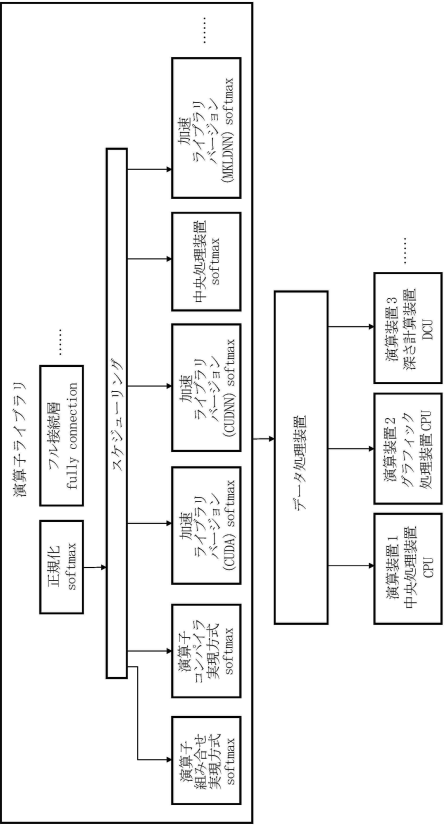
40

50

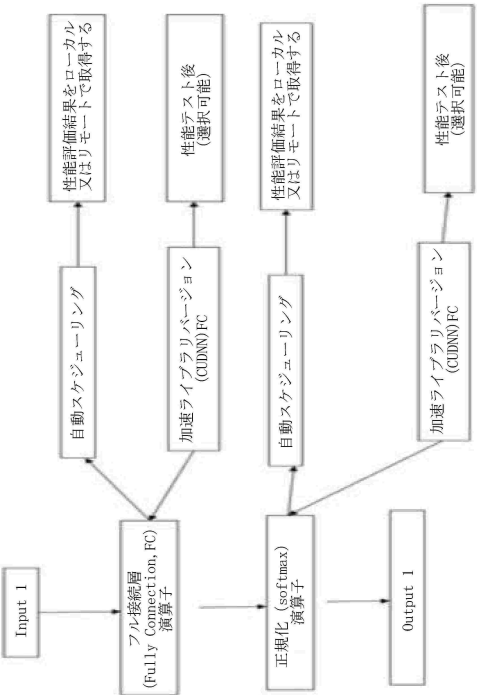
【図 9】



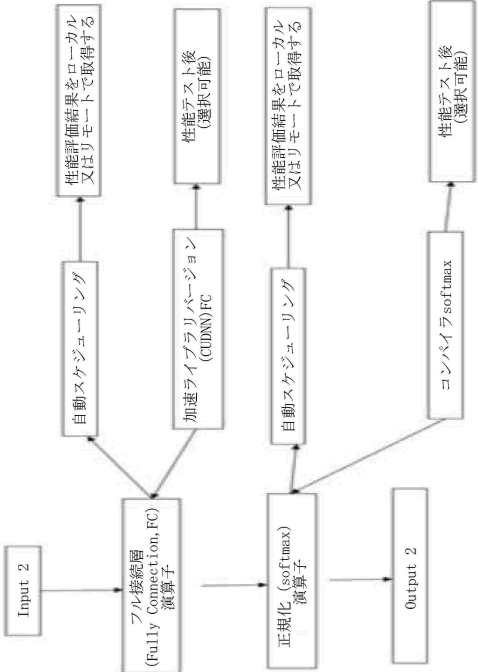
【図 10】



【図 11】



【図 12】



10

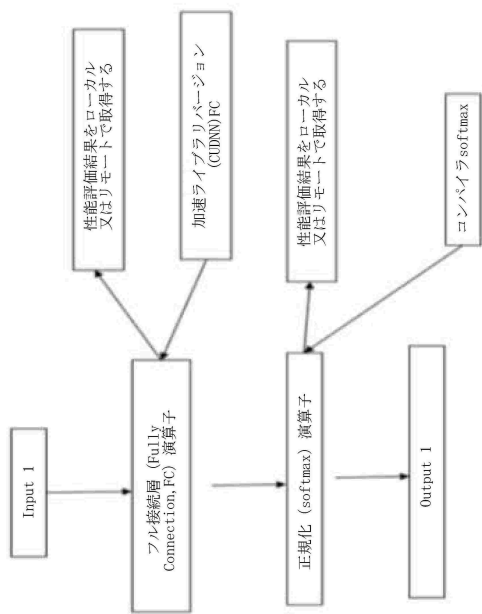
20

30

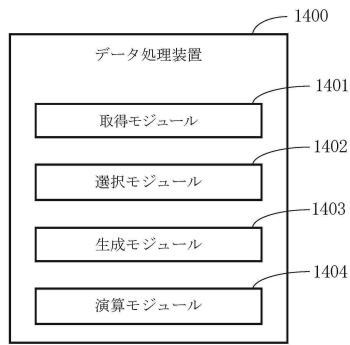
40

50

【図 1 3】

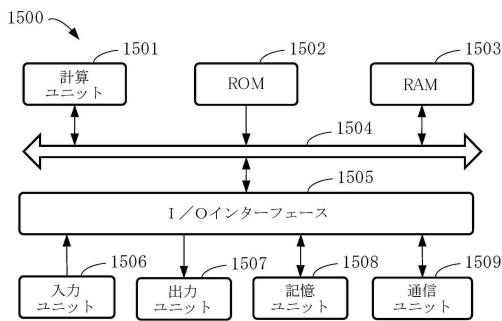


【図 1 4】



10

【図 1 5】



30

40

50

フロントページの続き

- 85, China
(74)代理人 100118913
弁理士 上田 邦生
(74)代理人 100142789
弁理士 柳 順一郎
(74)代理人 100201466
弁理士 竹内 邦彦
(72)発明者 ワン, ハイフェン
中華人民共和国 ペキン 100085, ハイディアン ディストリクト, シャンディ テンス ス
トリート, 10番, バイドウ キャンパス 2階
(72)発明者 フー, シャオガン
中華人民共和国 ペキン 100085, ハイディアン ディストリクト, シャンディ テンス ス
トリート, 10番, バイドウ キャンパス 2階
(72)発明者 ユ, ディアンハイ
中華人民共和国 ペキン 100085, ハイディアン ディストリクト, シャンディ テンス ス
トリート, 10番, バイドウ キャンパス 2階
(72)発明者 ラン, シャン
中華人民共和国 ペキン 100085, ハイディアン ディストリクト, シャンディ テンス ス
トリート, 10番, バイドウ キャンパス 2階
(72)発明者 マ, ヤンジュン
中華人民共和国 ペキン 100085, ハイディアン ディストリクト, シャンディ テンス ス
トリート, 10番, バイドウ キャンパス 2階
審査官 坂東 博司
(56)参考文献 国際公開第2021/057722 (WO, A1)
米国特許出願公開第2022/0121903 (US, A1)
中国特許出願公開第110766147 (CN, A)
中国特許出願公開第110956272 (CN, A)
中国特許出願公開第114492737 (CN, A)
米国特許出願公開第2021/0248136 (US, A1)
米国特許出願公開第2021/0174214 (US, A1)
米国特許出願公開第2019/0108559 (US, A1)
(58)調査した分野 (Int.Cl., DB名)
G06F 17/16
G06N 3/04